



## A new traffic signaling model based on graph and deep reinforcement learning

Erhan Turan<sup>1\*</sup>, Beşir Dandı<sup>2</sup>, Engin Avcı<sup>3</sup>

<sup>1</sup>Distance Education Application and Research Center, Ardahan University, 75100, Merkez, Ardahan, Türkiye

<sup>2</sup>Department of Mechatronics Engineering., Faculty of Engineering and Natural Sciences, İskenderun Technical University, 31200, İskenderun, Hatay, Türkiye

<sup>3</sup>Department of Software Engineering., Faculty of Technology, Firat University, 23100, Merkez, Elazığ, Türkiye

### Highlights:

- A new traffic signaling methodology based on graph and DRL
- Distributed signaling with phase and duration calculation
- Modeling real map and real traffic data in SUMO

### Keywords:

- Traffic signaling
- Deep reinforcement learning
- Max flow graph
- SUMO simulator

### Article Info:

Research Article

Received: 01.03.2023

Accepted: 30.12.2023

### DOI:

10.17341/gazimmfd.1257860

### Acknowledgement:

This study is supported by Firat University Scientific Research Projects Coordination Unit (FÜBAP) with project number ADEP.23.20.

### Correspondence:

Author: Erhan Turan

e-mail:

erhanturan@ardahan.edu.tr

phone: +90 542 399 6199

### Graphical/Tabular Abstract

In this study, a new traffic signaling methodology based on Graph and Deep Reinforcement Learning is proposed to provide solutions to traffic congestion at intersections. A distributed signaling system has been developed by combining the fast and accurate calculation ability of the Ford-Fulkerson graph algorithm with the dimensionality and generalization capabilities of DRL. The developed signaling system was modeled on SUMO using the real map model and real vehicle data. It was trained and tested on SUMO. The flow chart of the proposed method is given in Figure A.

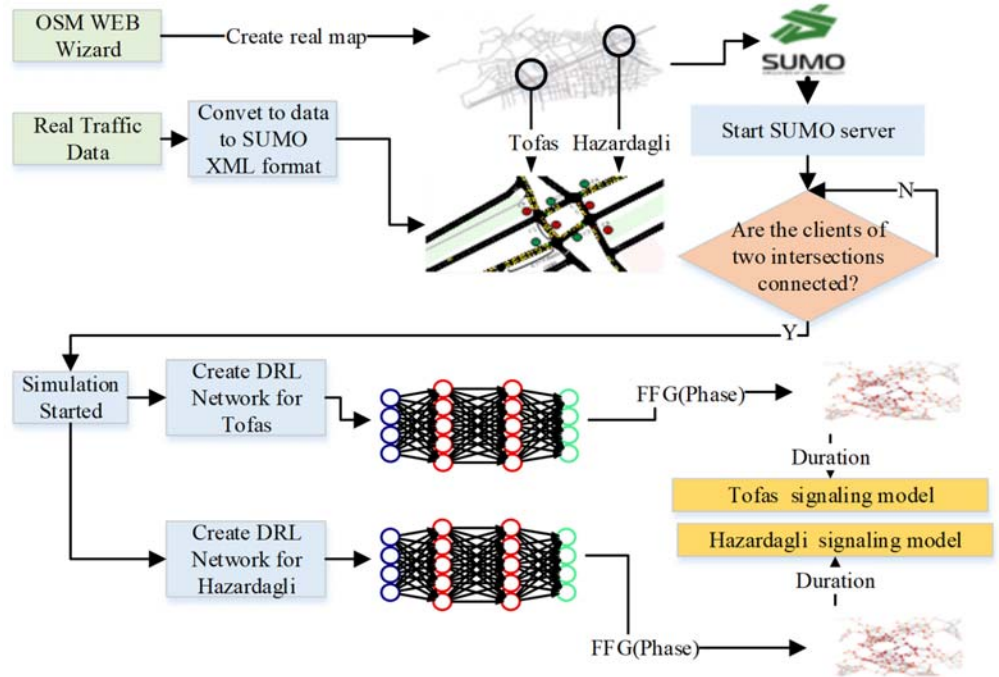


Figure A. Proposed Graph based DRL method

**Purpose:** It is aimed to increase the transit efficiency at intersections by reducing the average queue length and total waiting times at intersections.

**Theory and Methods:** Within the scope of the study, a new traffic signaling based on Graph and Deep Reinforcement Learning is proposed. This method calculates the phase sequence by DRL and the duration by the Ford Fulkerson graph method.

**Results:** The signaling model obtained as a result of training using the real map and real data reduced the queue length by 45% compared to the fixed time and sequential phase method.

**Conclusion:** In this study, a new signaling method has been developed based on distributed architecture without the need for a central management system, increasing signaling efficiency.



## Graf ve derin pekiştirme öğrenme tabanlı yeni bir trafik sinyalizasyon modeli

Erhan Turan<sup>1\*</sup>, Beşir Dandil<sup>2</sup>, Engin Avcı<sup>3</sup>

<sup>1</sup>Ardahan Üniversitesi, Uzaktan Eğitim Uygulama ve Araştırma Merkezi (UZEM), 75100, Merkez, Ardahan, Türkiye

<sup>2</sup>İskenderun Teknik Üniversitesi, Mühendislik ve Doğa Bilimleri Fakül., Mekatronik Müh., 31200, İskenderun, Hatay, Türkiye

<sup>3</sup>Fırat Üniversitesi, Teknoloji Fakültesi, Yazılım Mühendisliği., 23100, Merkez, Elazığ, Türkiye

### ÖNEÇİKANLAR

- Graf ve DRL tabanlı yeni bir trafik sinyalizasyon metodolojisi
- Faz ve süre hesaplamalı dağıtık sinyalizasyon
- Gerçek harita ve gerçek trafik verisinin SUMO'da modellenmesi

### Makale Bilgileri

Araştırma Makalesi

Geliş: 01.03.2023

Kabul: 30.12.2023

### DOI:

10.17341/gazimmfd.1257860

### Anahtar Kelimeler:

Trafik sinyalizasyon,  
derin pekiştirmeli öğrenme,  
maksimum akış graf,  
SUMO simülatörü

### ÖZ

Topolojik yapı ve kavşaktaki araçların bekleme süreleri, trafik sıkışıklığının genel nedenleri olarak gösterilir. Topolojik yapıdaki iyileştirmeler uzun ve maliyetli projeler sonucunda gerçekleştirildiğinden kavşak sinyalizasyon uygulamaları akıllı kentlerin vazgeçilmez uygulama alanı olmaktadır. Kavşak sinyalizasyon uygulamalarında kavşak bazında veya ağ genelinde, araçların birim zamanda maksimum akışını sağlamak için faz sırası ve süresi hesaplanır. Kavşak sinyalizasyon optimizasyonu birçok değişken veriden etkilenen, gerçek zamanlı bir gerçek dünya problemidir. Bu nedenle en verimli sinyalizasyon yöntemini geliştirmek için halen çok sayıda çalışma yürütülmektedir. Bu çalışmada ağ genelinde kavşak noktalarındaki bekleme sürelerini azaltmak için yeşil fazın sırasını ve süresini optimize eden bir yaklaşım önerilmiştir. Bu yaklaşım, gerçek dünya haritasındaki şehir kavşaklarını birebir ölçeğine göre gerçek zamanlı araç verileriyle birlikte SUMO simülatörüne aktararak geliştirilmiştir. Graf tabanlı faz süresi ve Derin Pekiştirmeli Öğrenmeye (Deep Reinforcement Learning-DRL) dayalı faz sırası tahminini birleştirerek GDRL adlı yeni bir sinyalizasyon yaklaşımı önerilmiştir. Bu çalışmada faz sırası DRL yöntemiyle hesaplanmaktadır. Faz süresi ise Ford-Fulkerson algoritmasının maksimum akış bulma yönteminden yola çıkılarak hesaplanır. GDRL yaklaşımı gerçek haritadaki ardışık kavşaklar üzerinde paralel çalıştırılarak ve gerçek veriler kullanılarak SUMO simülatöründe test edilmiştir. GDRL yaklaşımının, kavşaklardaki kuyruk uzunluğunu %44 oranla azaltarak, trafik sıkışıklığının çözümünde verimli sonuçlar ürettiği gözlemlenmiştir.

## A new traffic signaling model based on graph and deep reinforcement learning

### HIGHLIGHTS

- A new traffic signaling methodology based on graph and DRL
- Distributed signaling with phase and duration calculation
- Modeling real map and real traffic data in SUMO

### Article Info

Research Article

Received: 01.03.2023

Accepted: 30.12.2023

### DOI:

10.17341/gazimmfd.1257860

### Keywords:

Trafik sinyalizasyon,  
deep reinforcement learning,  
max flow graf,  
SUMO simulator

### ABSTRACT

Topological structure and waiting times of vehicles at the intersection are shown as common causes of traffic congestion. Since the improvements in the topological structure can be realized as a result of long and costly projects, intersection signaling applications become an indispensable application area of smart cities. In intersection signaling applications, phase sequence and duration are calculated to ensure the maximum flow of vehicles per unit time, on the basis of intersection or throughout the network. Junction signaling optimization is a real-time real-world problem that is affected by many variable data. Therefore, many studies are still being carried out to develop the most efficient signaling method. In this study, an approach that optimizes the order and duration of the green phase is proposed to reduce the waiting times at the junction points across the network. This approach has been developed by transferring real-time vehicle data to the SUMO simulator according to the exact scale of city intersections on the real-world map. A new signaling approach named GDRL is proposed by combining graph-based phase duration and Deep Reinforcement Learning-based phase sequence estimation. In this approach, the phase sequence is calculated by the DRL method. The phase duration is calculated based on the maximum flow-finding method of the Ford-Fulkerson algorithm. The GDRL approach has been tested in the SUMO simulator by running parallel on consecutive intersections on the real map and using real data. It has been observed that the GDRL approach efficiently solves traffic congestion by reducing the queue length at intersections by 44%.

\*Sorumlu Yazar/Yazarlar / Corresponding Author/Authors : \*erhanturan@ardahan.edu.tr, besir.dandil@iste.edu.tr, enginavci@firat.edu.tr /  
Tel: +90 542 399 6199

## 1. Giriş (Introduction)

### 1.1. Teknolojik Yaklaşım (Technological Approach)

Derin Pekiştirmeli Öğrenme (DRL) trafik sinyalizasyon modellerinin üretilmesinde yüksek doğruluk düzeyinde genelleme yapabilen yöntemlerin başında gelmektedir. Karmaşık trafik ağlarına kolayca uygulanabilmesi, kural tablosu veya denetimli eğitim setine ihtiyaç duymaması DRL modelinin geliştirilmesini kolaylaştırmaktadır. Çoğu DRL tabanlı sinyalizasyon modeli, sabit faz süresi kullanarak faz sırasını tahmin etmek için kullanılır. Sadece faz sırasını veya sadece faz süresini hesaplamak, araçların kavşak noktalarında gereksiz bekleme sürelerine neden olmaktadır. Yapılan sinyalizasyon çalışmalarının birçoğunda dağılım yöntemleriyle üretilmiş sentetik araç verileri kullanılmaktadır. Gün içerisindeki zamansal yoğunluk farklarından dolayı sentetik modeller gerçek uygulamalara aktarıldığında simülâtördeki verimliliklerine ulaşamamaktadır. Hızlıca artan kent nüfusu ve bireysel alanlara olan talebin artması kentlerde araç sayısındaki artışı hızlandırmıştır. Toplu taşımının iyileşmesi ve kullanımının artmasının yanında bireysel araç sayılarında yüksek artış oranlarına ulaşılmıştır. Bu durum sınırlı kapasiteli şehir trafik akışını yavaşlatarak kent yaşamındaki verimliliği azaltır. Araç sayısının artması trafikte oluşturduğu sıkışıklık nedeniyle kentsel bölgelerde yoğun karbonmonoksit (CO) gazının birikmesine neden olacaktır. CO gazı, günümüzün insan ve canlı yaşamını tehdit ederek fark edilmeyen zehir ve sessiz katil olarak adlandırılır [1]. CO gazının insanlarda etkisi incelendiğinde hipoksik aralanma nörolojik kırımlı ve hatta zamanla ölüme neden olabildiği, bitkilerde verimli ömrün kısalması, iklimsel olarak küresel ısınma ve ozon tabakasındaki incelmeye neden olduğu gözlemlenmiştir. Trafik sıkışıklığının ekonomik maliyeti ele alındığında yadsınamayacak kadar yüksek olduğu görülmektedir. Trafik sıkışıklığının Avrupa Birliğine maliyetinin yaklaşık GSYİH'nin %1'ine denk geldiği belirtilmektedir [2]. Amerika'da 2011 yılında trafik sıkışıklığından kaynaklı 24 milyon tonun üzerinde CO salındığı, 2014'te Amerikalıların fazladan 6.9 milyar saat seyahat etmesine ve 160 milyar dolarlık tıkanıklık maliyeti karşılığında fazladan 3.1 milyar galon yakıt tüketimine neden olduğunu belirtilmiştir [3]. INRIX 2019 ulaşım analitiğine göre, Amerika Birleşik Devletleri'nin en sıkışık bölgesi olan Boston'da yerel halk, işlerine ulaşabilmek için trafik sıkışıklığından kaynaklı yıllık ortalama 149 saat ve ortalama kişi başı 2205 dolar kayıp yaşamıştır [4]. Bu ekonomik kayıp ülke ekonomisine ise toplamda 4.1 milyar dolara mal olmuştur.

Trafikteki araçların bekleme süresini azaltmak için akıllı trafik sinyalizasyon sistemleri için geniş çapta çalışmalar yapılmıştır [5]. Akıllı trafik sinyalizasyonu, araçların trafikteki bekleme süresini en aza indirmeyi amaçlayan gerçek zamanlı bir dünya problemine çözüm olarak geliştirilmektedir. Trafik üzerinde elde edilen büyük veri ve yüksek kapasiteli işlem gücüne sahip olmamıza rağmen günümüzde halen basit kural tabanlı sinyalizasyon modelleri kullanılmaktadır. Gerçek zamanlı trafik akışına göre araç yönlendirmelerinin yapılması trafik sıkışıklığını ve CO gazının salınımını önemli ölçüde azaltacaktır. Bu nedenle yapay zeka tabanlı yaklaşımlar, makine öğrenme tabanlı yaklaşımlar, meta-sezgisel yaklaşımlar ve fonksiyon tabanlı yaklaşımlar giderek daha fazla araştırılmaktadır [6]. Araç yoğunlukları kamera, GPS, manyetik, mesafe ve lazer gibi sensörlerin bağlı olduğu IOT cihazlarıyla elde edilerek sensör füzyonu sayesinde araçların yönlendirmeleri sağlanmıştır [7].

Karmaşık olmayan trafik sinyalizasyon uygulamalarında kural tabanlı yapay zeka yöntemlerinin verimli olduğu çalışmalar yürütülmektedir. Pekiştirmeli Öğrenme (Reinforcement Learning-RL) geleneksel sinyalizasyon yöntemlerinden farklı olarak araç verilerine göre oluşturulan ödül tablosuna göre çalışmaktadır. Q değeri RL ajanının

eylemi seçme değeridir. RL modelinde amaç deneme yanılma yoluyla aksiyon seçme tablosu olan Q tablosundaki Q değerini değiştirerek ödül değerini maksimize etmektir [8]. Trafik ağ yapısının büyüklüğüne göre kural tablosu da büyür. Asimetrik kavşakların bulunduğu trafik ağlarında, Q tablolarının oluşturulması uygulama açısından uzun süre almaktadır. RL yöntemi trafik sinyalizasyonu uygulamasında umut verici olsa da, karmaşık trafik ağları, kural tablosunda olmayan istisnai durumlar açısından gelişimi kısıtlı bir yöntemdir [9].

DRL yöntemi, RL yönteminde kullanılan Q tablosunun yerine derin sinir ağı kullanılmaktadır. Bu sinir ağı sayesinde her ağ modeli için ayrı bir kural tablosu oluşturulmasına gerek kalmamaktadır [10]. DRL yöntemi ile model ödül değeri olarak bekleme süresi, kuyruk uzunluğu gibi çevresel verileri kullanabilir. Geleneksel pekiştirmeli öğrenmenin yavaş çalışması ve karmaşık ağ modellerine yanıt vermemesinden dolayı DRL yöntemi trafik sinyalizasyon sistemlerinde yaygın olarak kullanılan yöntemlerin başında gelmektedir.

Graf tabanlı maksimum akış algoritmaları, graf üzerinde başlangıç ve hedef düğüm arasında daima maksimum akışı hesaplar. Malezya, Sabah, Kota Kinabalu'da trafik ağının darboğazlarını belirlemek için maksimum akış minimum kesim teoremi olan Ford-Fulkerson Graf algoritması kullanılmıştır [11]. Bu çalışmada kullanılan Graf yöntemi, trafik sıkışıklığı sorununu en aza indirmek için darboğazların tahminini yaparak iyileştirme önerisinde bulunmuştur. Trafik ışık noktaları düğüm, yollar ise kenar kabul edilerek bütün trafik yapısı Graf yapısına dönüşür [12]. Graf yapısında araçların ana düğümünden hedef düğüme maksimum akış sağlanarak gidebilmesi için kenarların ağırlık değerlerinin derin sinir ağı, regresyon analizi gibi yöntemlerle belirlenmesi gerekmektedir.

### 1.2. Literatür Taraması (Literature Review)

Akıllı şehirlik kavramı, kaynak tüketimindeki verimliliği artırarak şehirlerin sürdürülebilirliğini amaçlamaktadır. Trafik sinyalizasyon optimizasyonu akıllı şehirliğin en önemli problemi olarak değerlendirilebilir. Fonksiyon tabanlı yöntemler trafik ışıklarını kontrol etmek için en ilkel yöntemler olarak kabul edilir. Bu yöntemde araçların kavşağa giriş ve çıkış sayısı, türü ve yönlenme oranlarına göre sabit bir fonksiyon üzerinden anlık olarak trafik fazı veya süresi hesaplanmaktadır [5]. Kavşaklardaki araç yoğunluğuna göre sinyal faz sırasını ve süresini belirlemek için fonksiyonel ve yapay zeka tabanlı yöntemler önerilmiştir. Maksimum basınç [13], yeşil dalga [14] ve Webster [15] adlı fonksiyon tabanlı yöntemler trafik sinyalizasyon sistemlerinde kullanılmaktadır.

Trafik sinyalizasyon sistemlerinde gerçek zamanlı uygulamalar geliştirmek için RL tabanlı yöntemler önerilmiştir. Durumun kuyruk uzunluğu, ödülün ise iki aksiyon arasındaki gecikme olarak tasarlandığı bir kavşağın kontrolü için RL algoritması sunulmuştur [16]. Tekli kavşağın yanı sıra çoklu kavşak sinyalizasyon kontrolünde de RL yöntemi kullanılmaktadır [17]. Trafik ağ yapısı genişledikçe RL yöntemindeki veri giriş sayısı ile birlikte hesaplama karmaşıklığı artmaktadır. Düşük, orta veya yüksek sıkışıklık seviyesine sahip ağlarda, özellik olarak faz sinyallerine göre ayrık araç verileri kullanılarak RL modelinin boyut problemine çözüm üreten bir model önerilmiştir [18]. RL'nin yavaş yakınsama ve uzun eğitim süresine yol açan yüksek durum eylemi karmaşıklığına çözüm olarak Growing Neural Gas öğrenme yapısının uygulanmasıyla eğitim süresi azaltılarak yüksek yakınsama oranı ve sistem kararlılığına yol açtığı gözlemlenmiştir [19]. Bu yöntem simülasyon üzerinde sinyalizasyon kavşakta değerlendirilerek, RL tabanlı optimizasyon yöntemleriyle karşılaştırılmıştır. Sonuçlar değerlendirildiğinde bu yöntemle, durum-

eylem karmaşıklığı azaltılarak, giriş verisi olan gerekli sayıda nöronun ön işleme ihtiyacı duymadan RL tabanlı yöntemi geliştirdiği görülmektedir. Yukarıda verilen çalışmalarda RL tabanlı sinyalizasyon yöntemleri, geleneksel yöntemlerden daha iyi sonuçlar üretse de dahili derin sinir ağı içermediğinden küresel maksimuma veya minimuma yakınsama açısından yeterince verimli değildir.

Derin sinir ağının ve GPU programlamanın gelişmesiyle birlikte RL'nin boyutsallık problemi aşarak yerine trafik sinyalizasyon uygulamalarında DRL yöntemi almıştır. Trafik sinyalizasyon sistemlerinde herhangi bir zamanda trafik hacmini tahmin etmek için derin sinir ağı tabanlı bir çalışma gerçekleştirilmiştir. Bu çalışmada derin sinir ağının fonksiyon tabanlı, rastgele yürüyüş ve destek vektör makinesine göre daha başarılı sonuçlar ürettiği görülmüştür [20]. DRL modeli eğitilirken gerçek verilerin yanı sıra dağılım yöntemleri kullanılarak sentetik veriler oluşturulabilir. Weibull dağılımını kullanarak dört kollu tek kavşak üzerinde, kuyruk uzunlukları ödül değerine göre eğitilen DRL modeliyle faz sırası ayarlanarak, trafik akışında verimlilik sağlandığı görülmüştür [21]. Boyutsallık probleminin DRL'de olmamasından dolayı farklı giriş verileri birleştirilerek daha iyi genelleme yapabilen modeller üzerine çalışmalar yapılmıştır. Kavşak kuyruk uzunlukları, kontrol ortamının nitelikleri, günün saati, haftanın günü, sinyal durumu ve kuyruk uzunlukları kullanılarak eğitilen DRL modelinde sabit zamanlı kavşağa göre %37 gecikme performansı sağlanmıştır [22]. Kapsamlı trafik verilerini kullanmanın aksine kısmen gözlemlenebilir veriler kullanarak trafik sinyali kontrolünü optimize etmek için derin takviyeli bir derin Q-öğrenme modelinin, maksimum basınç ve SOTL yöntemine göre %20'daha verimli olduğu görülmüştür [23]. Trafik sıklığını azaltmak için kavşak sinyalizasyon sistemlerinin yanı sıra rota belirleme çalışmaları da trafik sıklığına etki etmektedir. Bu duruma çözüm olarak bir aracın hedefe ulaşmak ve tıkanıklık yolundan kaçınmak için bir hat boyunca en optimal rotayı DRL yöntemiyle hesaplanmıştır [24]. DRL tabanlı birçok sinyalizasyon yöntemi önerilmiştir. IntelliLight adlı yaklaşım ödül değerini, kuyruk uzunluklarının toplamı, gecikmelerin toplamı ve yaklaşan tüm şeritlerdeki güncellenmiş bekleme süresinin toplamı dahil olmak üzere çoklu faktörlerin ağırlıklı toplamını alan DRL tabanlı sinyalizasyon yöntemidir [25]. FRAP adlı yaklaşım, farklı yol yapıları ve trafik koşulları için üstün genelleme sağlayabilen, faz rekabetinin sezgisel ilkesine dayanan RL tabanlı bir yöntemdir [26]. Yukarıdaki DRL tabanlı yöntemlerin çoğu, sinyalizasyon verimlilik performansında umut verici olsa da, bunlardan çok azı kavşak koordinasyonunu ve değişken faz süresini hesaba katmaktadır.

Trafik ağı içerisinde, kavşak sinyalizasyon sisteminde gerçekleştirilen eylem, doğrudan bağlantılı olduğu komşu kavşakları, dolaylı olarak ağ üzerindeki diğer kavşakların araç geçişlerini etkileyecektir. Bu nedenle, kavşaklar arasındaki geçiş ilişkileri doğru modellenirse, genel sinyalizasyon performansı optimum düzeye çekilebilir. CoLight adlı yaklaşım, iki kavşak arasındaki ofsetleri önceden hesaplayarak sinyal kontrolü için komşu kavşakların etkilerini birleştiren RL tabanlı bir yöntemdir [27].

Fonksiyon tabanlı yöntemler basit hesaplama ve işlem karmaşıklığı açısından karmaşık olmayan kavşak yapılarında başarılı sonuçlar üretebilmektedir. Fonksiyon tabanlı olan maksimum basınç yöntemi, kavşak kontrolü için fazların basıncını minimize ederek komşu kavşaklar için kuyruk uzunluğunu dengeleyerek aşırı yoğunluk riskini azaltmayı amaçlamaktadır [13]. Maksimum basınç yöntemi basit durum ve ödül tasarımı için RL tabanlı kontrol modellerinde kullanılmaktadır [28]. PressLight adlı yaklaşım, basınç değerine göre çalışan bazı trafik ağlarında araçların seyahat süresini önemli ölçüde azaltan etkili bir RL yöntemidir [9]. Maksimum basınç DRL ile verimli sonuçlar üretse de sadece şeritlerdeki araç sayısını dikkate alarak hız ve konum verilerini ihmal etmektedir.

Trafik sinyalizasyonu modellerinin verimlilik analizini ve trafik akışını izlemek için simülasyonlar vazgeçilmez uygulamalardır. Bu nedenle simülasyonlar, karmaşık araç ve yaya davranışlarını analiz etmek için gerçek dünya sistemlerinin veya süreçlerinin taklidi olarak trafik mühendisliğinin test ortamı olarak kullanılır [6]. CORSIM [29], MATSim [30], Paramics [31], SUMO [32-34], VISSIM [35], CityFlow [36] uygulamaları ticari ve bilimsel trafik sinyalizasyon sistemlerinin geliştirilmesinde en yaygın kullanılan trafik simülasyon çözümleridir. SUMO simülasyon uygulamasının açık kaynak olması, araç rotalamanın karar modellemesi, harici bir programlama dili, uygulama programlama ara yüzü, araç tipi, demiryolu modelleme, yaya modelleme, gerçek harita üzerinde ağ oluşturabilme yetenekleri belirtilerek DRL tabanlı bir trafik akış yönteminde kullanılmıştır [34]. SUMO simülasyon uygulaması üzerinde trafik yoğunluğu, yolculuk süresi, mesafesi, emisyon değerleri ve yeniden yönlendirme yetenekleri tespit edilerek yöntem araştırma çalışmaları yürütülmüştür [37]. Basit ve deneysel trafik ağları manuel oluşturulsa da gerçek trafik ağlarının oluşturulmasını kolaylaştırmak için alt uygulamalar geliştirilmiştir. SUMO üzerinde bulunan OSM Web Wizard ile gerçek harita verilerinin ağ, araç ve yönlendirme dosyaları otomatik şekilde oluşturularak, simülasyon üzerine aktarılır [38].

Graf teorisi trafik ağını temsil etmek ve uygun rotaları bulmak için sinyalizasyon ve yönlendirme çalışmalarında kullanılmaktadır. Zurich şehri için geliştirilen trafik sinyalizasyon yönteminde Dal ve Sınır graf algoritması kullanılarak hesaplama karmaşıklığı azaltılmıştır [39]. Ardışık kavşaklar bir akış ağına dönüştürülerek, bağlantı yollarının yoğunluğu akış ağındaki kapasiteye göre değiştirildiği bir maksimum akış tabanlı çalışma yapılmıştır [40]. Maksimum akış tabanlı bu çalışmada, her kavşak için gelen yolların giden yollara karşı akışı belirlenerek yeşil sinyal süresi hesaplanır. Yüksek yoğunluklu trafik akışlarında graf tabanlı bu yaklaşımın, sabit zamanlı yaklaşıma göre daha yüksek akış sağladığı görülmüştür.

İncelediğimiz literatüre göre çalışmamız, trafik sinyalizasyonundaki verimliliği daha iyi optimize edebilmek için ardışık kavşaklarda gerçek harita ve araç verisi kullanılarak, DRL ve Ford Fulkerson Graf yöntemini birleştirerek faz sırası ve süresini hesaplayabilen ilk çalışmadır.

### 1.3. Katkı (Contribution)

Ağ düzeyinde kavşak sinyalizasyonu yapmak için bu makale, DRL ile Ford Fulkerson tabanlı Graf yöntemini birleştirerek faz sırası ve süresini optimize etmektedir. Makale aşağıda verilen üç önemli katkı sunmaktadır.

1. Faz sırasının DRL modeliyle süresinin ise Graf tabanlı yöntemle hesaplanarak birleşmesiyle yeni bir sinyalizasyon modeli geliştirilmiştir. DRL modeli derin sinir ağı ile öğrenme gerçekleştirildiğinden yüksek kapasiteli GPU hızından yararlanır. Önerilen GDRL modelinde sadece faz sırası için DRL kullanılırken faz süresinin hesaplanmasında fonksiyon tabanlı Ford Fulkerson Graf yöntemi kullanılmaktadır.
2. Ardışık kavşaklar paralel çalışan GDRL modelleriyle kontrol edilmektedir. Merkezi bir model çıktısıyla değil her kavşağın GDRL modeli kendi araç verisiyle üretilmektedir. Bu durum gerçek dünyadaki trafik ağ yapısına en uygun modeli optimize etme imkanı sağlamaktadır.
3. Önerilen GDRL modeli, gerçek araç verileri gerçek harita üzerine aktarılarak SUMO simülatörü üzerinde test edilmiştir. Literatürdeki gerçek araç verilerini kullanan DRL modellerine göre yüksek başarımlı sağlamıştır.

Bu makalenin geri kalanı şu şekilde düzenlenmiştir. Bölüm 2'de, önerilen GDRL yaklaşımının yöntem, veri ve simülasyon ayrıntıları

verilmiştir. Bölüm 3'te, gerçek araç ve harita verilerine ilişkin simülasyon sonuçları sunulmuştur. Son olarak, Bölüm 4'te makalenin sonuç kısmı verilerek genel değerlendirme yapılmıştır.

## 2. GDRL Sinyalizasyon Metodolojisi (GDRL Signaling Methodology)

Bu bölümde, yeni bir trafik sinyalizasyon modeli olan GDRL yaklaşımının faz sırası ve süresinin belirlenmesindeki sürecin akış şeması ayrıntılarıyla verilerek anlatılmıştır.

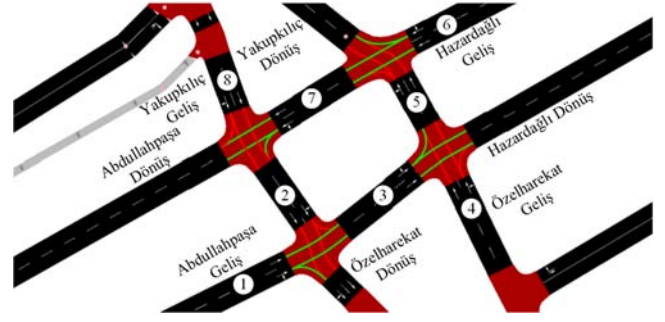
### 2.1. Gerçek harita üzerinde kavşak modelleme (Intersection modeling on real map)

Trafik terminolojisinde kavşağa giren şeritler gelen şerit, ayrılan şeritler ise çıkan şerit olarak adlandırılır. Kavşağa giren her şeridin çıkış yapabileceği bir veya daha fazla çıkış şeridi olabilir. Her girişin eşit sayıda çıkışının olabileceği gibi farklı sayıda çıkışı olan heterojen yapıdaki kavşaklar da mevcuttur. GDRL yaklaşımında Elazığ ilindeki Tofaş ve Hazardağlı kavşakları, gerçek harita verileri olarak SUMO simülatörü üzerine Şekil 1'de verildiği gibi aktarılmıştır.

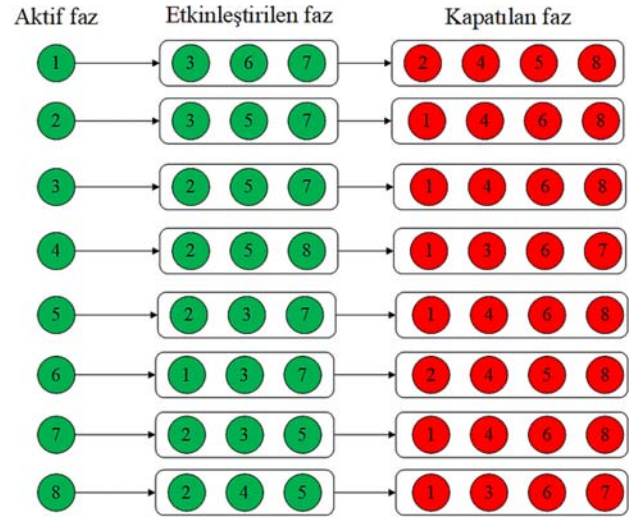
Şekil 1'de verilen Tofaş ve Hazardağlı kavşağı benzer simetrik giriş çıkış şeritlerine sahiptir. Ardışık kavşak olmalarından dolayı çıkışları değerinin girişini etkilemektedir. Faz, trafik ışıklarında araç geçiş sıralarını belirleyen, kavşakta birbirlerini kesmeyen şerit geçişlerinin döngüsel kombinasyonları olarak tanımlanır. Yeşilin izin verilen hareket anlamına geldiği kırmızının ise hareketin yasaklandığı sinyal olarak işleme alınmaktadır. Tofaş kavşağının, SUMO simülatörü üzerindeki faz geçiş izinleri Şekil 2'de verilmiştir.

Şekil 2'de verilen Tofaş kavşağının 1'den 8'e kadar faz adlandırmaları yapılmıştır. 1,4,6,8 fazları ana faz 2,3,5,7 fazları ise ara faz olarak değerlendirilir. Bu faz geçişleri birbirlerini karşılıklı tetikleyerek aktifleşirler. Kavşağa gelen ana fazlar ile kavşak içindeki ara fazlar birbirlerini kesemezler. Şekil 2'de verilen Tofaş kavşağının aktif fazının etkinleştirdiği akış fazları Şekil 3'te verilmiştir.

Şekil 3'te verilen akış fazlarında aktif faz, DRL modeliyle belirlenen fazdır. Bu faz aktifleştğinde sadece kendi fazını değil karşılıklı ve akışın devamlılığını sağlayan fazları da etkinleştirir. Etkileşen fazların kesişmemesi için aktif ve etkinleştirilen fazların haricindeki diğer fazlar kapatılmaktadır.



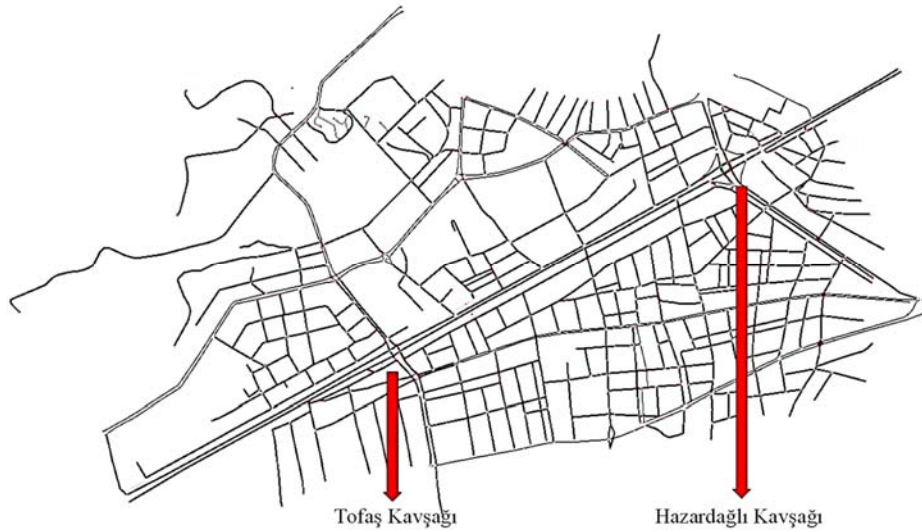
Şekil 2. Tofaş kavşağı faz geçişleri (Tofaş junction phase transitions)



Şekil 3. Akış fazları (Flow phases)

### 2.2. GDRL Sinyalizasyon Mimarisi (GDRL Signaling Architecture)

Graf ağı ve derin sinir ağı yapılarının birlikte kullanıldığı trafik sinyalizasyon metodolojisi önerilmiştir [12]. Önerilen bu yaklaşımda graf ağı üzerinde derin sinir ağı kullanılarak araç akış yönelimlerinin tahmini yapılmaktadır. Bu tahmini yapabilmek için tamamen denetimli veri setine ihtiyaç duyulmaktadır. Bu çalışma genel bir



Şekil 1. GDRL modelinde kullanılan Elazığ ili gerçek harita modeli (Real map model of Elazığ province used in the GDRL model)

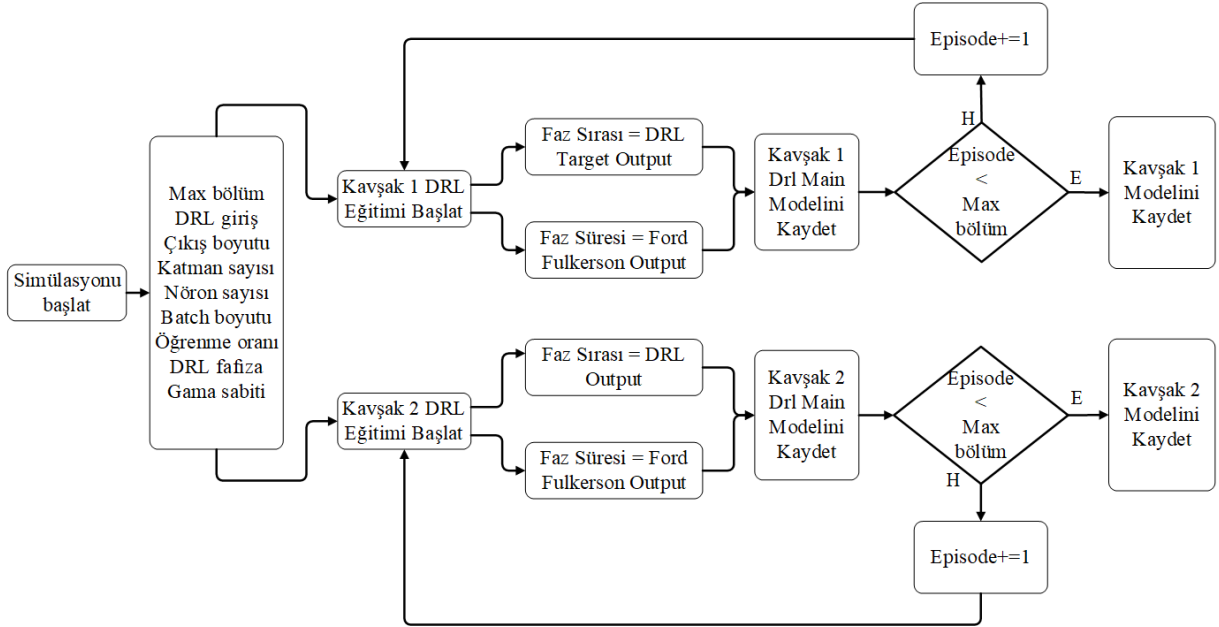
metodoloji yaklaşım olarak değerlendirilir. GDRL yaklaşımı ise denetimli veri setine ihtiyaç duymadan yarı denetimli ve sentetik veriler üzerinde çalışabilmektedir. GDRL yaklaşımının bu çalışmaya katkısı genel teknolojik yöntemleri matematiksel modele dökerek kod, uygulama, eğitim ve test işlemlerinin yapılarak doğrulanmasıdır. Literatürde incelenen çalışmalara göre DRL ağının faz sırasında, maksimum akış tabanlı grafın ise sinyalizasyon süresinin hesaplanmasında kullanılması GDRL'nin bu alandaki yeni bir metodoloji olduğunu göstermektedir. GDRL yaklaşımı DRL ve Ford Fulkeron Graf yöntemini birleştirerek, faz ve süre optimizasyonu ile trafik sinyalizasyon yöntemlerine yeni bir bakış açısı kazandırmıştır. DRL ile faz sırası hesaplanırken aynı anda düşük hesaplama maliyeti olan Graf yöntemiyle de faz süresi hesaplanarak sinyalizasyon yürütülür. Önerilen GDRL modeli simülasyon üzerinde iki ayrı kavşak için paralel bir şekilde yürütülmüştür. Her kavşak için GDRL sinyalizasyon modeli paralel proselerde eğitilerek model şeklinde kaydedilir. Kavşaklar için kayıtlanan GDRL modelleri daha sonra test aşamasında sinyalizasyon için kullanılmaktadır. GDRL modelinin

simülasyon üzerinde çalışmasını gösteren akış şeması Şekil 4'te verilmiştir.

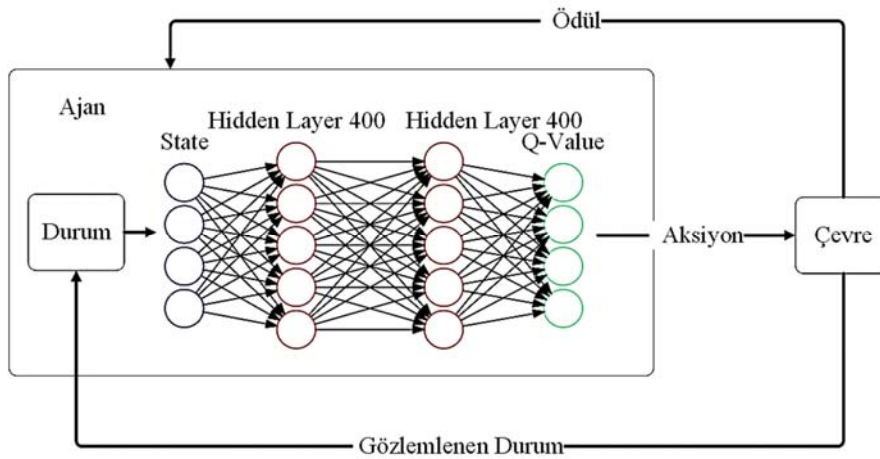
### 2.3. GDRL Faz Tasarımı (GDRL Phase Design)

Bu alt bölüm GDRL yönteminde faz sırasını belirlemek için kullanılan DRL yönteminin, durum, aksiyon ve ödül tasarımı detaylandırılır. Her kavşak için ayrı ayrı eğitilen DRL modeli ödül değerini artırmak için DRL ağının çıkışı olan Q değerini güncellemektedir. Ödül değeri araçların anlık ve bir önceki fazdaki bekleme süresine göre hesaplanmaktadır. Bu sayede, DRL yönteminin hedefi araçların bekleme süresini minimuma indirerek kavşak yoğunluğunu azaltmaktır. DRL yönteminin öğrenme adımları Şekil 5'te verilmiştir.

- **Durum:** Bu çalışmada trafik ağı olarak gerçek harita ve araç verisi kullanılmıştır. Çaydaçıra ve Hazardağlı kavşaklarındaki ara ve ana fazlar üzerinde bulunan araçların toplamı durum değerini verir.

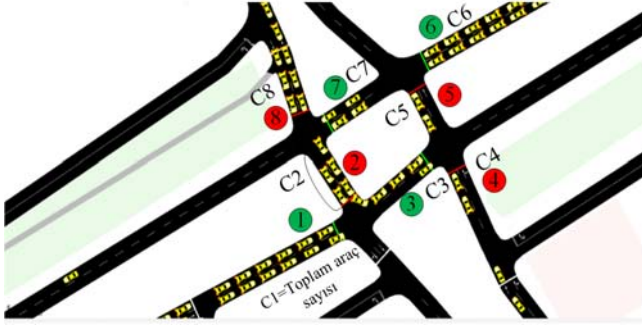


Şekil 4. GDRL modeli akış şeması (GDRL model flow chart)



Şekil 5. DRL modeli öğrenme adımları (DRL model learning steps)

Durum bilgisi her faz değişikliğinde tekrar hesaplanır. Her faz geçişine göre araçların toplanarak durum değerini oluşturduğu yapı Şekil 6'da verilmiştir.



Şekil 6. Durum değeri oluşturma (Create a status value)

Şekil 6'da verilen C1, C2, C3, C4, C5, C6, C7, C8 değerleri her faz için kendi şeridinde bulunan toplam araç sayılarını ifade eder. Bu araç sayıları Şekil 5'te verilen DRL modelinin durum değerini oluşturur. 1x8 boyutunda bir dizi olarak DRL ağına giriş olarak verilir.

- **Aksiyon (Eylem):** Önerdiğimiz GDRL yöntemindeki DRL modelinin temel hedefi, kavşaktaki araç bekleme sürelerini minimuma indirebilmek için en iyi fazı seçebilmektir. Her fazın işletme süresi bittiğinde DRL modeli yeni fazı seçmek zorundadır. Şekil 5'te verilen aksiyon, ajanın seçtiği fazdır. DRL modelinin faz çıkışı 8 aksiyona göre tasarlanmıştır. DRL modeli her tahmin işleminde 8 fazdan en iyi olanı seçerek sinyalizasyonu yürütür.
- **Ödül:** DRL yönteminde ödül değeri Q değeri olan aksiyon seçiminde doğrudan etkilidir. Yüksek ödül değeri daha doğru aksiyon seçimi anlamına gelmektedir. Bu çalışmada ödül değeri,

araçların bir önceki fazla anlık faz arasında kümülatif bekleme sürelerinin farkı olarak Eş. 1'de verilmiştir.

$$\text{reward} = \text{old\_total\_wait} - \text{current\_total\_wait} \quad (1)$$

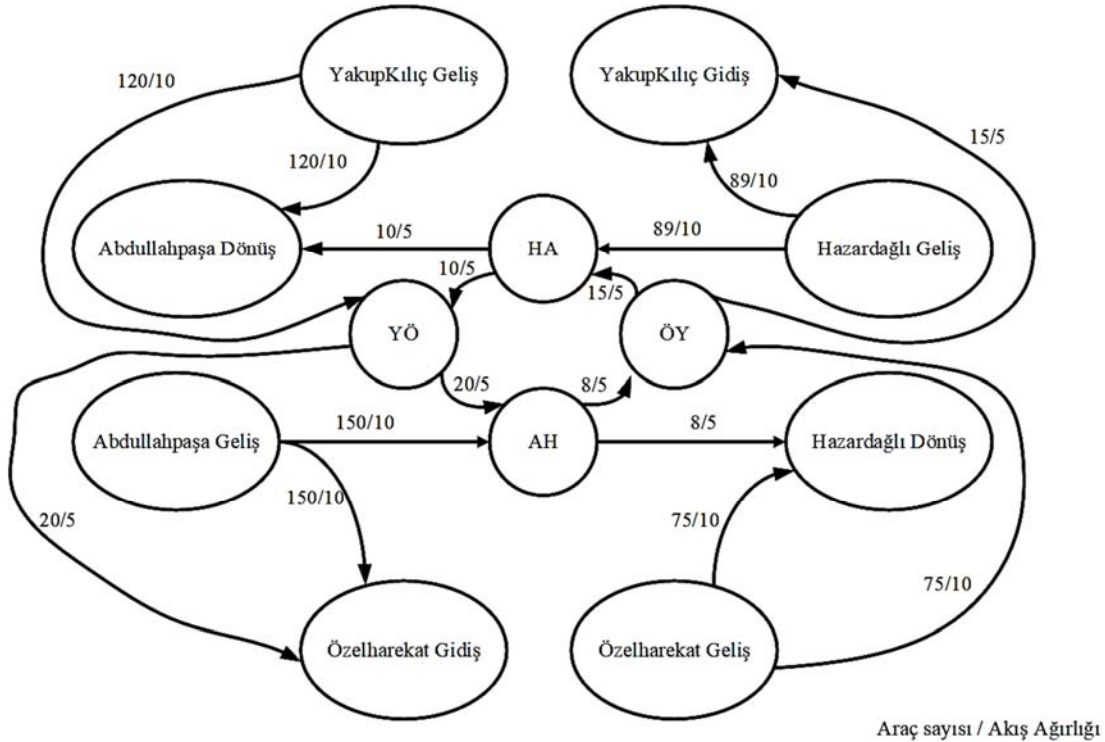
DRL yönteminde ödül değerine göre ağırlık çıkışı verisi olan Q değeri güncellenir. Ajan Q değerinin güncellenmesini Bellman denklemine göre yapmaktadır. Eski Q değerinin yerini güncel Q değeri olarak ağırlık yeniden eğitilir.  $Current\_q[action]$  değeri yeni Q değerini, reward ödülü, gamma Bellman sabitini,  $q\_s\_a\_d[i]$  ise  $i$ . durumun Q değerlerini temsil eder. Yeni Q değerini hesaplamak için kullanılan Bellman denklemi Eş. 2'de verilmiştir.

$$\text{Current\_q}[action] = \text{reward} + \text{gamma} * \text{np. amax}(q\_s\_a\_d[i]) \quad (2)$$

Bellman denklemine göre model parametrelerini doğru sıklıkla güncellemek büyük önem arz eder. Model ağırlıklarının çok sıklıkla güncellenmesi modelin yavaş öğrenmesine, geç güncellenmesi erken yakınsamaya giderek yavaş öğrenmeye neden olur [21]. Çalışmamızda güncelleme başlangıcı için en az 400 verilik bir minimum hafıza doluluğu aramaktayız. 400 veri en az 400 faz geçişi demektir. Her kavşak için en az 400 faz geçişi tamamlandığında DRL yönteminin ana ağı yeni Q değerlerini hesaplayarak güncellenmeye başlar ve yeniden eğitilir.

#### 2.4. GDRL Faz Süresi Tasarımı (GDRL Phase Duration Design)

GDRL yönteminde geleneksel trafik sinyalizasyonundan farklı olarak, faz sırasıyla birlikte değişken sürede sinyalizasyon süresi hesaplanabilmektedir. Yaklaşımımızda minimum ve maksimum yeşil süresi arasında fazın aktifleştirdiği hat boyunca graf akışına göre sinyalizasyon süre optimizasyonu yapılır. Faz süresi hesaplanırken faz geçişinin aktifleştirdiği şeritlerin akış ağırlıkları da önemlidir. Bu nedenle araçların şeritlerdeki sayısı ve akış ağırlıklarını dikkate alan Graf tabanlı Ford Fulkerson algoritmasıyla faz süresi hesaplanır. GDRL yönteminin eğitim ve test işlemlerinde kullanılan Tofaş kavşağının graf yapısı Şekil 7'de verilmiştir.



Şekil 7. Tofaş kavşağı graf yapısı (Tofaş intersection graph structure)

Şekil 7’de Tofaş kavşağına ait örnek araç sayısına göre hazırlanan graf yapısı verilmiştir. Bu graftaki araç sayısı şeritte bekleyen araç sayısını, kapasite ise şeridin akış ağırlığını ifade etmektedir. DRL yöntemi, faz sırasının seçileceği eylemi belirledikten sonra bu fazın Ford Fulkerson Graf (FFG) yöntemine göre faz süresi belirlenerek sinyalizasyonu yürütmektedir. Sinyalizasyon süresi hesaplanırken  $t_{min}$  minimum yeşil süresini,  $t_{max}$  ise maksimum yeşil süresini ifade eder. Aktif fazda araç varsa yeşil süresi daima  $t_{min}$  ve  $t_{max}$  arasında normalize edilir. Seçilen aksiyonun şeridinde araç yoksa  $t=0$  olarak sinyalizasyon işletildikten sonra DRL modeli çevreyi gözlemleyerek yeni aksiyonu seçmek için model üzerinde yeniden tahmin yapar. Her seçilen aksiyon Şekil 3’te verildiği gibi 3 tane daha fazı aktifleştirir. Seçilen aksiyon (Aktif Faz-AF), seçilen aksiyonun etkinleştirdiği fazlar ise (Etkinleştirilen Faz-EF) olarak ifade edilerek Şekil 7’de verilen Graf yapısına göre faz süresinin hesaplanması Eş. 3’te verilmiştir.

$$\text{Faz süresi} = \begin{cases} \{0 \mid \sum \text{araç}_{AF} = 0\} \\ \text{oran} = \frac{\text{FFG}_{\text{AraçSaysı}}(\text{AF}) + \text{FFG}_{\text{AraçSaysı}}(\text{EF})}{\text{FFG}_{\text{AkışAğırlığı}}(\text{AF}) + \text{FFG}_{\text{AkışAğırlığı}}(\text{EF})} \\ \text{oran} > 1 \rightarrow \text{oran} = 1 \\ \{t_{min} + \text{oran} * (t_{max} - t_{min})\} \end{cases} \quad (3)$$

### 2.5. GDRL Model Öğrenme Süreci (GDRL Model Learning Process)

Bu bölümde, GDRL yönteminin öğrenme aşamaları detaylandırılarak verilmiştir. DRL yöntemi yüksek genelleme performansı, boyutsallık avantajı ve karmaşık problem çözümünde yüksek başarıma sahiptir. DRL yöntemi eğitim aşamasında karar vermek için ana ağı ve eğitim sonucunda modeli kaydetmek için ise hedef ağı kullanır. Periyodik olarak belirli bir çevrimden sonra ana ağıdaki veriler hedef ağıya kopyalanır. GDRL yaklaşımımız gerçek veri ve sentetik verilere uygun olarak tasarlanmıştır. Trafik sinyalizasyon sistemlerinde yaygın olarak kullanılan RL ve DRL yöntemlerinden farklı olarak GDRL yöntemi, faz sırası ve süresini aynı anda hesaplayarak birlikte yürütebilmektedir. Faz süresini hesaplarken ayrı bir DRL yerine düşük hesaplama maliyeti olan Graf tabanlı Ford Fulkerson’dan yararlanmaktadır. Eğitim işleminde eşit süreye ayrılmış ardışık zaman dilimine göre gerçek araç verileri kullanılmıştır. GDRL yöntemi model oluştururken her faz değişiminde DRL ajanı, kavşaklardaki araç sayılarının durumu alır. Faz sırasını ana ağ ile faz süresini de Graf ile belirledikten sonra simülasyonu yürütür. Eşit sürelerle bölünmüş her simülasyon, DRL *epoch* sayısına göre hafızadan *batch size* kadar örnek alarak hedef ağını eğitir. GDRL yönteminin eğitim işlemi Tablo 1’deki algoritmada detaylandırılarak verilmiştir.

### 3. Deneysel Sonuçlar (Experimental Results)

Çalışma kapsamında önerilen GDRL yöntemi GTX1070 GPU, i5 6600 CPU ve 8 GB RAM ve Windows 10 işletim sistemi üzerinde uygulanmıştır. Python üzerinde Tensorflow-Gpu, Keras, Matplotlib ve SUMOLib kütüphaneleri kullanılarak geliştirilmiştir. Tensorflow-Gpu tabanlı DRL modeli, Cuda 11.6. versiyonu ve Nvidia 512.15 versiyonlu sürücü üzerinde yürütülmektedir.

#### 3.1. SUMO Üzerinde Gerçek Harita Modeli Oluşturma (Creating a Real Map Model on SUMO)

SUMO trafik simülatörü karmaşık trafik yapılarını modellemek için geliştirilmiştir. Karmaşık trafik yapısını simülasyona aktarabilmek için trafik ağı, araç ve yaya rotaları ve SUMO konfigürasyon dosyalarının oluşturulması gerekmektedir. Bu simülasyon dosyaları oluşturularak yapılandırma klasörünün içine atılması gerekmektedir. SUMO’da bir trafik ağına düğümler kavşakları, kenarlar ise kavşakları birbirine bağlayan yolları temsil eder. SUMO’da trafik ağını oluşturmak için Nedit uygulaması kullanılmaktadır. Rota

dosyası trafik ağındaki kenar verileri kullanılarak araç ve yayaların başlangıç ve hedef noktası arasında zaman, hız ivme gibi verilerine göre oluşturulur. Genellikle araç türleri ve yayalara göre ayrı rota dosyaları oluşturulur. Simülasyon üzerinde olan hareketli nesnelerin tamamı rota dosyasında tanımlanmaktadır. SUMO yapılandırma dosyası içerisinde, ağ ve rota dosyalarının adları ve dosya yol bilgilerini barındırır. SUMO simülatör uygulaması simülasyonu başlattığında çalıştırdığı ilk dosya yapılandırma dosyasıdır. Bu dosyadaki dosya isimleri ve yolları yanlış yüklenirse simülasyon başlatılamaz. Nedit üzerinden oluşturulan trafik ağına dosya adı “trafikAğAdı.net.xml”, rota dosyası “binekarac.rou.xml” ve son olarak konfigürasyon dosyası “trafikAğAdı.sumocfg” yapısında olmalıdır. SUMO konfigürasyon dosyasının XML yapısı Şekil 8’de verilmiştir.

SUMO simülatörüne aktarılan ağ dosyası ve araç rota dosyasının oluşturulduğu Nedit uygulaması Şekil 9’da verilmiştir.

Şekil 9’da verilen trafik ağ yapısı XML tabanlı bir dosya olarak oluşturulur. Bu ağ yapısı üzerinde araç ve yayalar eklenerek rota dosyası oluşturulur. OSM web sihirbazı, açık kaynak harita olan OSM üzerinde SUMO simülasyonunda için gerçek harita ağları oluşturmak için kullanılan web tabanlı bir uygulamadır. Bu uygulama ile harita üzerinden belirli bir alan seçildikten sonra bu alanda işletilecek araç türü ve sayısı belirlenmektedir. Seçilen alana göre ağ dosyası, araç türü ve yoğunluğuna göre rasgele yönlendirilen rota dosyası oluşturularak simülasyon başlatılır. OSM web sihirbazı Python tabanlı uygulamadır. OSM web sihirbazı üzerinden trafik ağına alanı, araç türleri ve yoğunlukları belirlenerek Şekil 10’da verildiği gibi gerçek harita simülasyonu oluşturulmaktadır.

Şekil 10’da verilen simülasyon uygulama aşamasında her araç için rasgele başlangıç ve hedef noktası seçilir. Geçişli Trafik Faktörü (Thorough Traffic Factor), simülasyon içerisindeki araçların hedef noktalarının simülasyon sınırı olabileceği olasılığıdır. Bu değer araç sayısını arttırmazken, araçların simülasyonda geçirdiği yol uzunluk olasılığını ve buna bağlı olarak yoğunluğunu artırır. Araç sayı sabiti (count) değeri simülasyondaki araç tipine göre simülasyona eklenecek olan araç sayısı sabitidir. Simülasyona eklenecek toplam araç sayısı Eş. 4’te verilmiştir.

$$\text{Araç sayısı} = \text{Toplam kenar uzunluğu(km)} * \text{şerit sayısı} * \text{count} \quad (4)$$

Şekil 10’daki OSM web sihirbazı üzerinden seçilen Tofaş ve Hazardağlı kavşaklarının bulunduğu alan Şekil 9’da verilen Nedit uygulaması üzerinde düzenlenerek simülasyona uygun hale getirilmektedir. Yan yollar, kavşağa giren yollar belirli bir mesafeden sonra silinerek simülasyona fazladan yük olması engellenir. OSM web sihirbazı üzerinden düzenlenen haritanın SUMO simülatöründeki yapısı Şekil 11’de verilmiştir.

Bir sinyalizasyon uygulamasının SUMO simülatörü üzerinde koşturulması için uygulama ve SUMO arasında bağlantı yapılması gerekmektedir. SUMO geliştiricileri bu bağlantının yapılabilmesi için Python, C++, C# ve Java programlama dilleri için TRACI (Traffic Control A Interface) adlı bir bağlantı kütüphanesi geliştirmişlerdir [46]. TRACI kütüphanesi SUMO üzerinde simülasyon başlatmak için TCP tabanlı sunucu ve istemci protokölüne göre çalışır. SUMO simülatörünün sunucu ve istemci uygulamaları, aynı port üzerinden simülatöre bağlanarak işlemlerini yürütmektedir [47]. SUMO sunucusu başlatıldıktan sonra sunucu, başlatıldığı porttan istemcilerin bağlantısını dinler. İstemcilerin bağlantısı yapıldıktan sonra simülasyon başlatılır. Simülasyonun tamamlanması için öncelikle istemcilerin sunucu bağlantılarını kapatması gerekmektedir. Bütün istemciler sunucuyla olan bağlantısını kapattıktan sonra sunucu kendi bağlantısını sonlandırarak simülasyonu bitirir. Simülatör ekranı canlı



**Tablo 1.** GDRL yöntemi eğitim işlemi (GDRL method training process)**Giriş verileri:**

- I.**  $E$  bölüm sayısı (Episode)
- II.**  $SS$ , Simülasyon adımı;
- III.**  $Q$ , action-value fonksiyonu(Bellman denklemi);
- IV.**  $S$ , çevresel durum
- V.**  $M$ , Hafıza boyutu
- VI.**  $\epsilon$ , rastgele bir eylem seçme olasılığı;
- VII.**  $BS$ , batch boyutu;
- VIII.**  $NS$ , state boyutu
- IX.**  $NA$ , aksiyon sayısı
- X.**  $NL$ , katman sayısı;
- XI.**  $WL$ , hücre sayısı;
- XII.**  $LR$ , öğrenme oranı;
- XIII.**  $EP$ , çevrim (epoch)
- XIV.**  $\gamma$ , gama-indirme faktörü
- XV.**  $M$ , hafıza boyutu
- XVI.**  $T_{min}$ , minimum yeşil süresi
- XVII.**  $T_{max}$ , maksimum yeşil süresi
- XVIII.**  $FW$ , akış ağırlıkları

**Eğitim İşlemi:**

```
DRLModel=DRLAğıOluştur(BS,NS,NA,NL,WL,LR)
m = 0;
```

```
for i ← 1 to E do
  j = 0;
  S = getState
  while j < SS do
    if Random(0, 1) ≤ ε then
      aksiyon = SelectRandomAction();
    else
      aksiyon = argmaxQ(DRL(S));
    fazSure = GFFSureHespla(S, Tmin, Tmax, FW)
    SimülasyonYürütme(aksiyon, fazSure)
    Snext = getState;
    ödül = beklemeSuresi(S) - beklemeSuresi(Snext)
    j = j + fazSure;
    Hafızakayı(S, aksiyon, ödül, Snext) in M;

  for k ← 1 to EP do
    batch = hafızadanRastgeleÖrnekOku(BS);
    QS = DRLModel.predict_batch(batch[0])
    QSnext = DRLModel.predict_batch(batch[3])
    for a, b ← 1 to enumerate batch do
      S, aksiyon, ödül, Snext = b[0], b[1], b[2], b[3]
      current_Q = QS[a]
      current_Q[aksiyon] = ödül + gamma * argmaxQ(QSnext[a])
      x[a] = S
      y[a] = current_Q
    DRLModel.train(x,y)
```



```
sumo_config.sumocfg - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
<?xml version="1.0" encoding="UTF-8"?>
<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://sumo.dlr.de/xsd/sumoConfiguration.xsd">
  <input>
    <net-file value="environment.net.xml"/>
    <route-files value="episode_routes.rou.xml"/>
  </input>

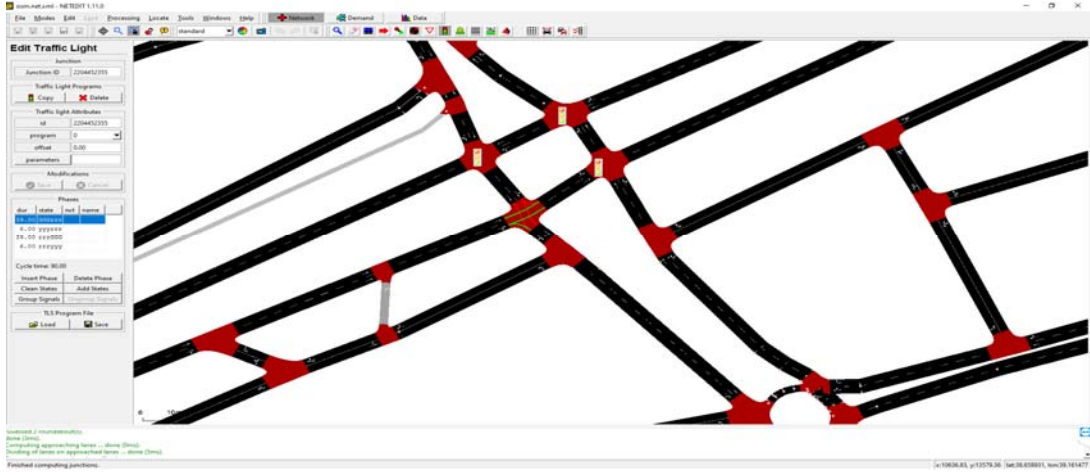
  <time>
    <begin value="0"/>
  </time>

  <processing>
    <time-to-teleport value="1"/>
  </processing>
```

**Şekil 8.** SUMO konfigürasyon dosyası (SUMO configuration file)

bir şekilde çalışarak simülasyon yürütülür. Bu durum sonucu işlemcisini zorlayabildiği ve manuel start butonuna basılma

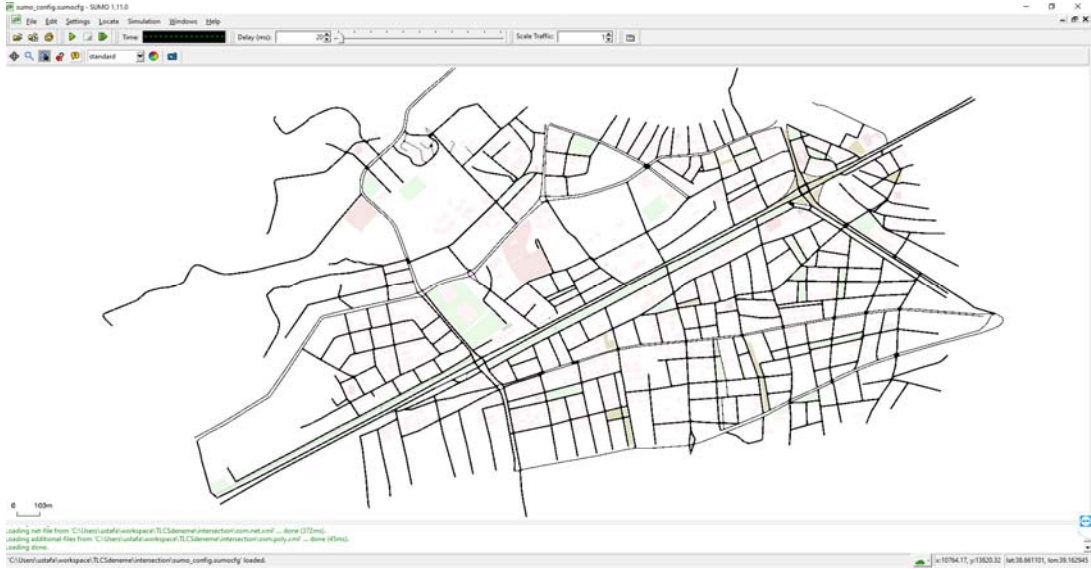
zorunluluğu olduğu için deneysel trafik model geliştirmelerinde arayüz kapatılarak komut ile otomatik bir şekilde simülasyon



Şekil 9. Netedit uygulaması (Netedit application)



Şekil 10. OSM web sihirbazı (OSM web wizard)



Şekil 11. SUMO simütörü (SUMO simulator)

bölmeleri yürütülmektedir. Sunucuya bağlanabilecek istemci sayısı, sunucu başlatılmadan önce “num-clients” komutu ile belirtilir. Bütün

istemciler simülasyon yürütmek istedikleri her komutun ardından “simulationStep” komutunu çağırmak zorundadırlar. Sinyalizasyon

yeşil, sarı sürelerinin işletilmesi için ayrı ayrı hesaplandıkları değer kadar “simulationStep” komutu çalıştırılır. Bu komut yürütülmeden simülasyon ilerlememektedir. Tüm araçların simülasyondan çıktığı “getMinExpectedNumber” adlı TRACI komutuyla algılanmaktadır. Sunucu ve istemciler simülasyondaki bütün araçların dolaşımını tamamladıklarını ve simülasyonda olmadıklarını algıladığında, kapatma komutu göndererek simülasyonu sonlandırmaktadırlar. Simülasyonun tamamen kapanması için bütün istemcilerin kapatma komutu göndermesi gerekmektedir. TRACI kütüphanesinin SUMO ile bağlantı akış şeması Şekil 12’de verilmiştir.

TRACI ile simülasyon üzerindeki hareketli ve hareketsiz nesnelerin bilgileri okunabilmektedir. Trafik simülasyonunda değişken değerlere sahip nesnelere ise TRACI üzerinden güncellenebilir. Trafik nesnelere okuma komutlarında araç tipi, sayısı, hareket rotası ve yaya sayı bilgileri alınabilmektedir. Sabit nesne olarak simülasyonda eklenen detektör verileri alınabilir. Kavşaktaki araç sayıları, şeritteki araç sayıları ve kenardaki araç sayıları alınarak işlenebilmektedir. Altyapı nesnelere olarak trafik ışıkları, otobüs durakları, şarj istasyonları, park alanları gibi veriler alınarak araç yönlendirmelerinde kullanılabilir. Trafik durum değiştirme komutları olarak araç konumları, rotaları, türü ve yaya durumu değiştirilerek simülasyonda yürütülebilmektedir. Trafik ışıklarının değerleri değiştirilerek sinyalizasyon uygulamaları yürütülebilmektedir.

### 3.2. Deneysel Ayarlar (Experimental Settings)

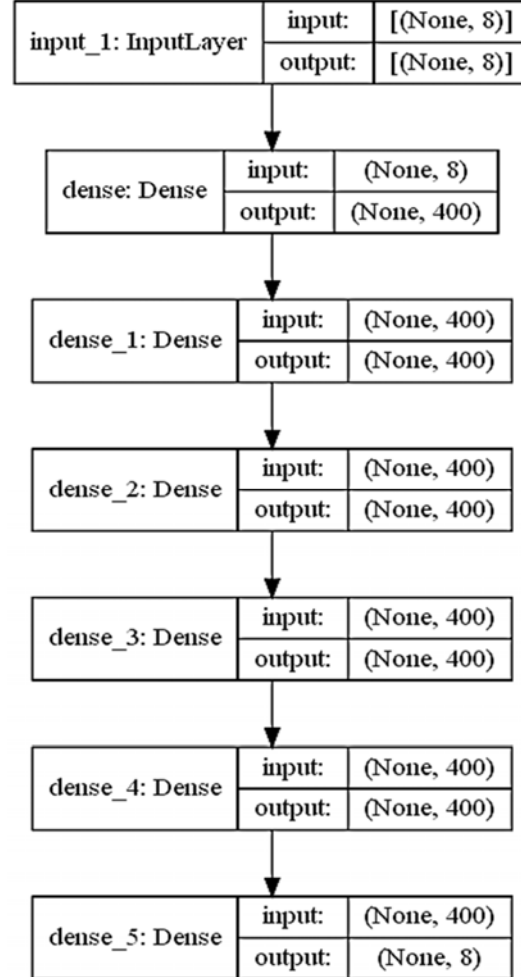
GDRL yöntemi gerçek harita kullanılarak iki kavşakta iki şeride sahip dört ana girişten oluşan ağ üzerinde geliştirilmiştir. Bu kavşaklar göbek alanına sahip olduğundan sinyalizasyon sistemine sahip olan ikişer şeritlik dört adet ara yola sahiptir. Simülasyon sırasında, trafik ağının her bir kavşağının dört ana ve dört ara yolu tek sinyalizasyon fazına sahiptir. Modelimizin sekiz şeritten alınan araç verileri giriş değerini, sekiz sinyal fazı ise çıkış değerini vermektedir. Modelimizin ara katman sayısı 4, nöron sayısı ise 400 olarak belirlenmiştir. Faz sırasını belirlemek için kullandığımız derin ağ mimarisini Şekil 13’te verilmiştir.

DRL modeli eğitilirken simülasyon bölüm sayısı 50, hafıza boyutu 50000, batch boyutu 400, öğrenme oranı 0.001, çevrim sayısı 400, gama-indirime faktörü 0.75, minimum yeşil süresi 14 adım, maksimum yeşil süresi 28 adım ve ana fazların akış ağırlıkları 10, ara fazların akış ağırlıkları 5 olarak alınmıştır.

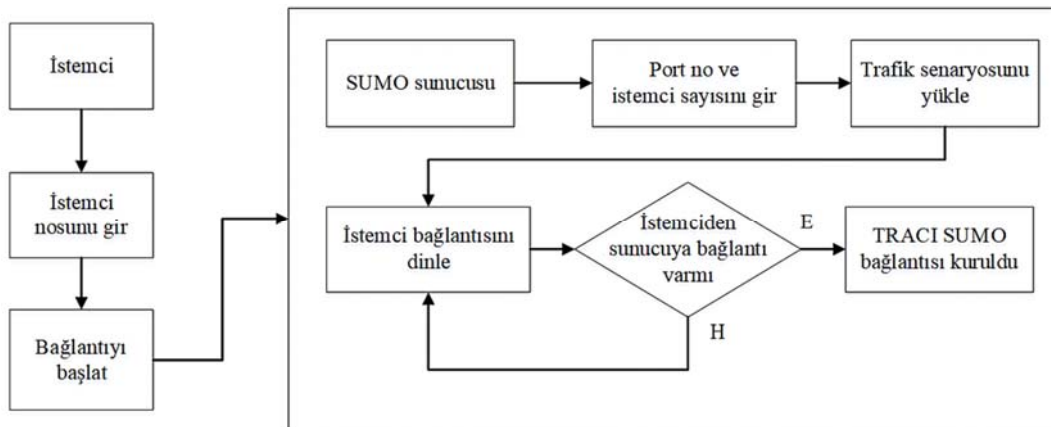
### 3.3. Veri Setinin Oluşturulması (Creating The Data Set)

Çalışma kapsamında Elazığ ilindeki Tofaş ve Hazardağlı kavşaklarının iki günlük araç verileri işlenerek simülasyona

aktarılmıştır. İki günlük veri 100 eşit parçaya bölünerek eğitim ve test için ayrılmıştır. Eğitim işleminde ilk 50 bölümün verisi, test işleminde ise sonraki 30 bölümün verisi kullanılmıştır. Son 20 bölümlük veri ise optimizasyon için ayrılmıştır. Gerçek veriler, faz geçişlerinde kavşağa gelen araç verilerini içermektedir. Tofaş kavşağı için kavşağa gelen araç sayılarını ve faz zamanını gösteren veri örneği Tablo 2’de verilmiştir.



Şekil 13. DRL modeli ağ mimarisini (DRL model network architecture)



Şekil 12. TRACI SUMO bağlantı şeması (TRACI SUMO connection diagram)

Tablo 1'in son satırı incelendiğinde 29/07/2021 tarihi saat 16:08:53'te 'Yakup Kılıç' tarafından Tofaş kavşağına 5 araç, 'Özel Harekat' girişinden ise 6 araç giriş yaptığı görülmektedir. Bu fazın yeşil süresi 11 saniye, bir sonraki fazın ise 26 saniyedir. Sonraki 'Yakup Kılıç' ve 'Özel Harekat' fazı, sarı fazlarının da yanmasıyla dördüncü satırda görüldüğü gibi 16:09:50 süresinde aktifleşmiştir. Bu veri gerçek hayatta 16:08:53-16:09:50 zamanı arasında, 'Yakup Kılıç' tarafındaki yoldan 7 araç, 'Özel Harekat' tarafındaki yoldan ise 3 araç geldiğini göstermektedir. Bu işlem kademeli olarak Tofaş ve Hazardağlı kavşağındaki fazlara uygulanarak otomatik bir şekilde XML tabanlı simülasyon verisi oluşturulmuştur. Gerçek verilerin simülasyon üzerinde çalışabilmesi için oluşturulan XML dosyasının, yönlendirme ve araçlarının tanımlandığı veri formatı Şekil 14'te verilmiştir.

Şekil 14'te örnek bir XML dosyasının örnek verileri bulunmaktadır. XML içerisinde 'route' yönlendirme etiketini, 'vehicle' ise araç tagını göstermektedir. Örnek olarak üçüncü satırda route tagı için id="WB1" ifadesi kullanılmıştır. Bu adlandırma Western-Between olarak çözümlenebilir. Gerçek hayatta ise Western 'Abdullahpaşa Geliş'B1 ise 'Hazardağlı Gidiş' olarak işlem görür. Edge değerleri ise simülasyon üzerindeki bağlantıyı oluşturan yolların adlarıdır. Onuncu satır incelendiğinde vehicle tagı içerisinde id="W\_N1\_5" olarak görülmektedir. Bu değer aracın Western (Abdullahpaşa Geliş)'den gelerek, N1 (Yakup Kılıç Gidiş) yönüne giden beşinci araç olduğunu söylemektedir. Ayrıca depart değeri simülasyona giriş süresini ve departSpeed ise maksimum hızını vermektedir. Simülasyon XML veri dosyasında araçlar süre sıralı olmak zorundadır. Simülasyona giren araçların XML dosyasında sıralanması için SUMO'nun "route.py" uygulaması kullanılmaktadır. GDRL modelinin eğitildiği gerçek araç

verilerinin kavşaklara olan talep akışları eş zamanlı olarak Şekil 15 ve Şekil 16'da verilmiştir.

Şekil 15 ve Şekil 16'daki x eksenini, ölçümün yapıldığı adımı temsil etmektedir. Bu şekillerdeki y eksenini ise x eksenine karşılık gelen adımdaki kavşağına, gelen bütün yollardaki araç sayısının toplamını temsil eder. Her iki grafiğin başlangıç ve bitiş zamanları aynıdır. Adım sayılarının farklı olmasının nedeni her iki kavşakta yeşil süreleri ve kavşağın veri okuma sürelerinin farklı olmasıdır.

### 3.4. Ortalama Kuyruk Uzunluklarının Karşılaştırılması (Comparison of Average Queue Lengths)

GDRL yönteminde DRL modelinin ödül tasarımı, aracın şeritteki birikmiş bekleme zamanları toplanarak hesaplanmıştır. Ödül değerleri için kuyruk uzunluğu ve maksimum basınç yönteminde yaygın olarak kullanılmaktadır. DRL yönteminde ödül verisi için basınç değeri kullanan özelleştirilmiş yöntemlere PressLight [9] ve IPDALight [41] örnek gösterilebilir. Maksimum basınç hesaplama yöntemi Şekil 17'de verilmiştir.

Maksimum basınç ve yoğunluğa göre seçilen bir sinyalizasyona sahip DRL yaklaşımında en az sayıda araca sahip şeritleri için adaletsiz araç bekleme sürelerine neden olabilir. Bu durum DRL yöntemi kullanan sinyalizasyon yaklaşımlarında olasılıksal Epsilon ( $\epsilon$ ) değeriyle aşılmaktadır. Epsilon değeri, bölüm sırası (*episode*) ve toplam bölüm sayısına (*total\_episode*) bağlı olarak Eş. 5'teki gibi hesaplanmaktadır.

$$\epsilon = 1.0 - \frac{\text{episode}}{\text{total\_episode}} \quad (5)$$

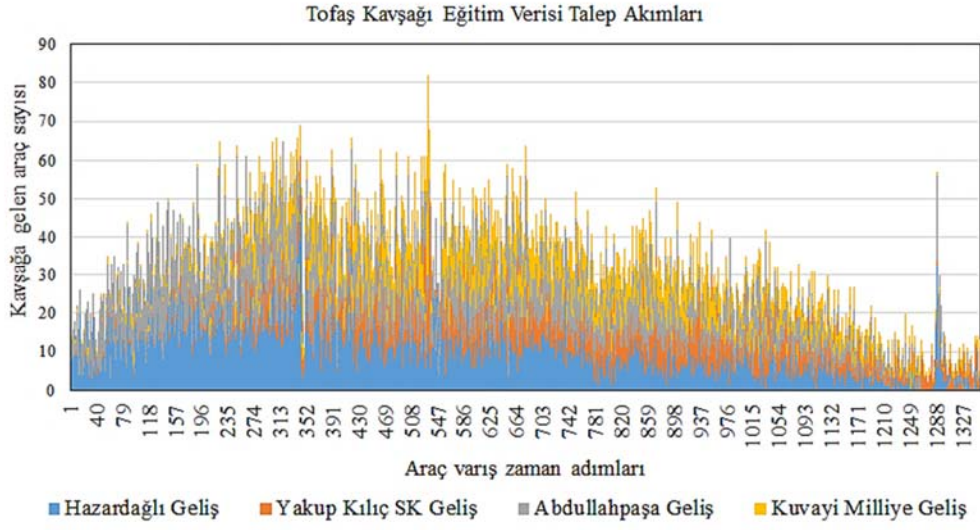
**Tablo 2.** Tofaş kavşağı gerçek araç veri örneği (Tofaş intersection real vehicle data example)

Tarih	Faz	Yön (Taşıt Sayısı)
29/07/2021 16:11:38	Yakup Kılıç SK ve Özel Harekat Geliş	Yakup Kılıç SK Geliş (9) Özel Harekat Geliş (8)
29/07/2021 16:10:59	Hazardağlı ve Abdullahpaşa Geliş	Hazardağlı Geliş (24) Abdullahpaşa Geliş (18)
29/07/2021 16:10:40	Yakup Kılıç SK ve Özel Harekat Geliş	Yakup Kılıç SK Geliş (9) Özel Harekat Geliş (5)
29/07/2021 16:10:09	Hazardağlı ve Abdullahpaşa Geliş	Hazardağlı Geliş (12) Abdullahpaşa Geliş (7)
29/07/2021 16:09:50	Yakup Kılıç SK ve Özel Harekat Geliş	Yakup Kılıç SK Geliş (7) Özel Harekat Geliş (3)
29/07/2021 16:09:12	Hazardağlı ve Abdullahpaşa Geliş	Hazardağlı Geliş (24) Abdullahpaşa Geliş (14)
29/07/2021 16:08:53	Yakup Kılıç SK ve Özel Harekat Geliş	Yakup Kılıç SK Geliş (5) Özel Harekat Geliş (6)

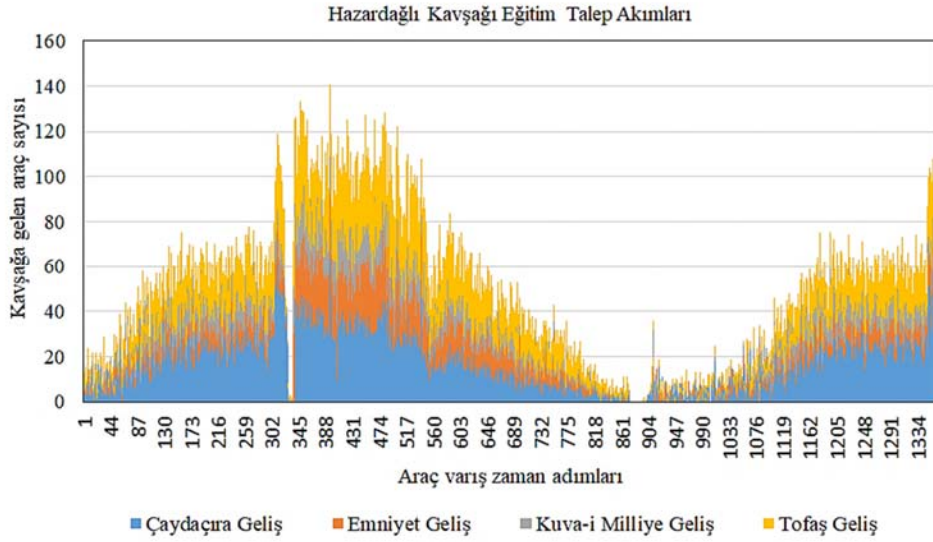
```

*episode_routes.rou.xml - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
<routes>
  <vType accel="1.0" decel="4.5" id="standard_car" length="5.0" minGap="2.5" maxSpeed="25" sigma="0.5" />
  <route id="W_N" edges="W2TL TL2N"/>
  <route id="W_E" edges="W2TL TL2E"/>
  <route id="W_S" edges="W2TL TL2S"/>
  <route id="N_W" edges="N2TL TL2W"/>
  <route id="N_E" edges="N2TL TL2E"/>
  <route id="N_S" edges="N2TL TL2S"/>
  <route id="E_W" edges="E2TL TL2W"/>
  <route id="E_N" edges="E2TL TL2N"/>
  <route id="E_S" edges="E2TL TL2S"/>
  <route id="S_W" edges="S2TL TL2W"/>
  <route id="S_N" edges="S2TL TL2N"/>
  <route id="S_E" edges="S2TL TL2E"/>
  <vehicle id="S_N_0" type="standard_car" route="S_N" depart="66.0" departLane="random" departSpeed="10" />
  <vehicle id="E_W_1" type="standard_car" route="E_W" depart="78.0" departLane="random" departSpeed="10" />
  <vehicle id="W_E_2" type="standard_car" route="W_E" depart="123.0" departLane="random" departSpeed="10" />
  <vehicle id="W_E_3" type="standard_car" route="W_E" depart="124.0" departLane="random" departSpeed="10" />
  <vehicle id="E_W_4" type="standard_car" route="E_W" depart="132.0" departLane="random" departSpeed="10" />
  <vehicle id="E_W_5" type="standard_car" route="E_W" depart="142.0" departLane="random" departSpeed="10" />
  <vehicle id="N_W_6" type="standard_car" route="N_W" depart="145.0" departLane="random" departSpeed="10" />
  <vehicle id="S_N_7" type="standard_car" route="S_N" depart="146.0" departLane="random" departSpeed="10" />
</routes>
  
```

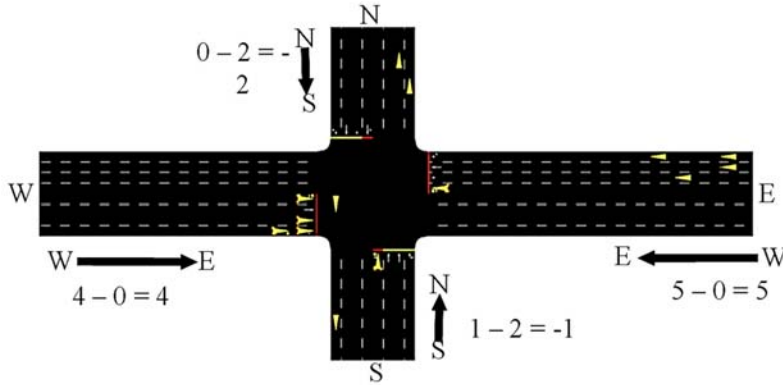
**Şekil 14.** Simülasyon XML veri yapısı (Simulation XML data structure)



Şekil 15. Tofaş kavşağı eğitim verisi talep akışları (Tofaş intersection training data request flows)

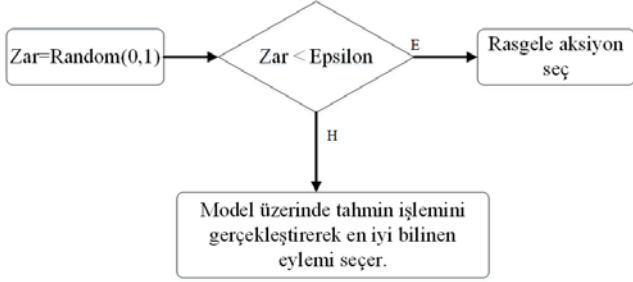


Şekil 16. Hazardağlı kavşağı eğitim verisi talep akışları (Hazardağlı intersection training data request flows)



Şekil 17. Maksimum basınç hesaplama (Maximum pressure calculation)

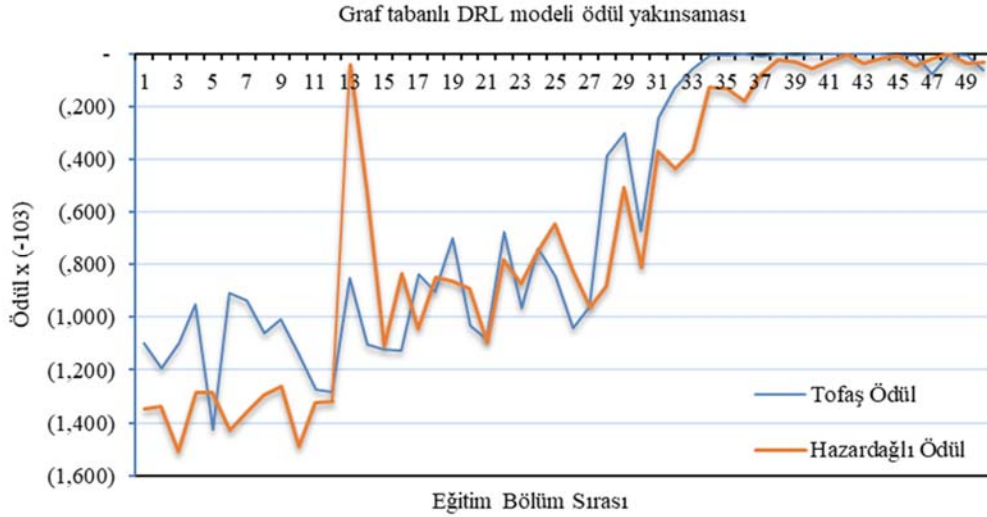
Faz seçimleri için daima aşırı değerler yerine olasılıksal fazlar seçilerek ödül değerinin maksimize edilmesi istenir. DRL modeli için kullanılan aksiyon seçme yöntemi Şekil 18’de verilmiştir.



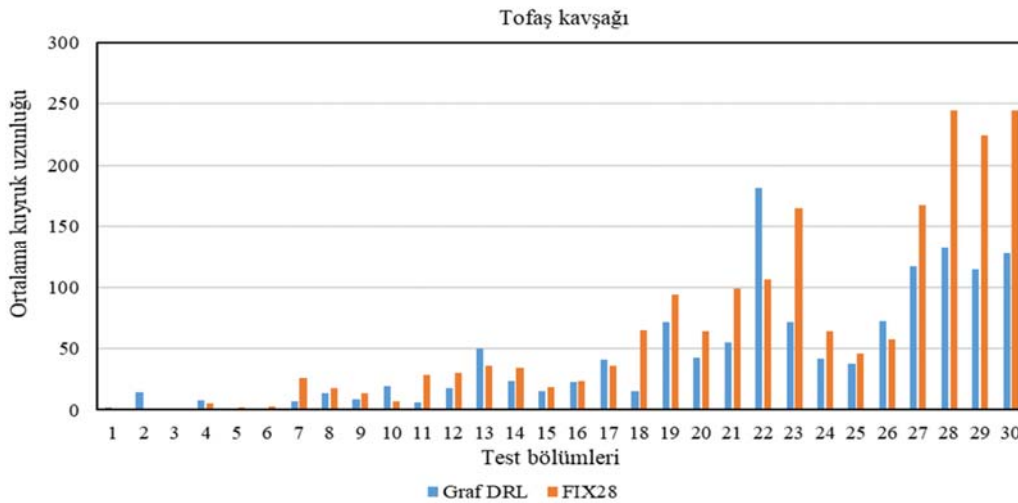
Şekil 18. DRL eğitim aksiyon seçme adımları (DRL training action selection steps)

GDRL yaklaşımı simülâtör üzerinde eğitilirken bölüm epoch sayısı kadar hafızasından rasgele batch boyutu kadar veri olarak ana ağına

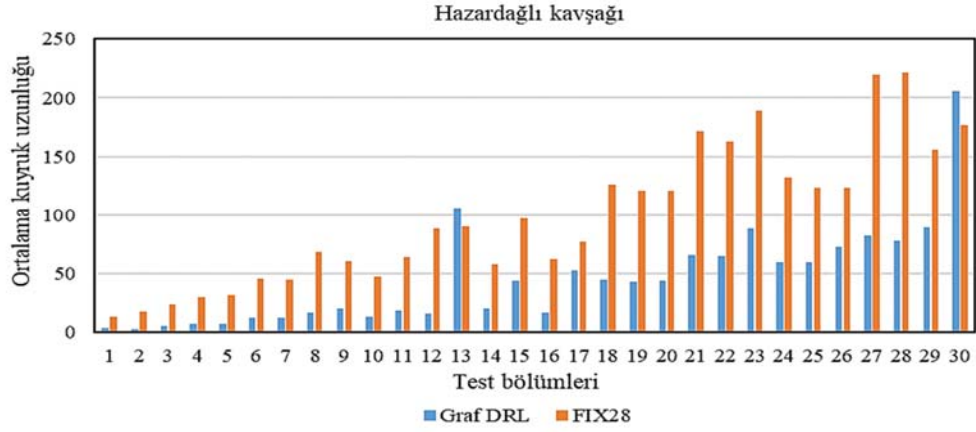
günceller. Bu ağ sonucu ödül değerini arttırmak için tasarlanmıştır. GDRL yöntemi eğitilirken Tofaş ve Hazardağlı kavşaklarında gerçek verilere göre oluşan ödül değerleri Şekil 19’da verilmiştir. Tofaş ve Hazardağlı kavşaklarında her faz süresi 28 adım işletilecek şekilde ayarlanarak FIX28 sabit süre ve sıra fazlı yöntem ile karşılaştırma yapılmıştır. GDRL yaklaşımımıza göre üretilen modelin Tofaş ve Hazardağlı kavşaklarının kuyruk uzunluklarına göre otuz bölümlük test sonuçları sırasıyla Şekil 20 ve Şekil 21’de verilmiştir. Doğunluk derecesi kavşağa gelen yol veya genel olarak bütün yollardaki birim zamanda geçen maksimum araç sayısı olarak ifade edilir. Saat başına düşen araç sayısı, binek araç birimi (Passenger car unit-PCU) olarak ifade edilir. Doğunluk derecesi  $V/C$  olarak hesaplanmaktadır [48]. Birim zamanda geçen araç sayısı genel kavşak için kavşağın kapasitesini geçemez. Kavşağa gelen yolun akışında ise yolun kapasitesini geçemez. Bu nedenle doğunluk derecesi 0-1 aralığında normalize değerlerden oluşmaktadır. Doğunluk derecesinin bire yaklaşması, kavşağın verimli sinyalizasyon yürüterek daha fazla araç akışının sağlandığını göstermektedir. Şekil 20’deki Tofaş ve Hazardağlı kavşaklarına ait sonuçların son beş bölümüne ait doğunluk dereceleri Şekil 22’de verilmiştir.



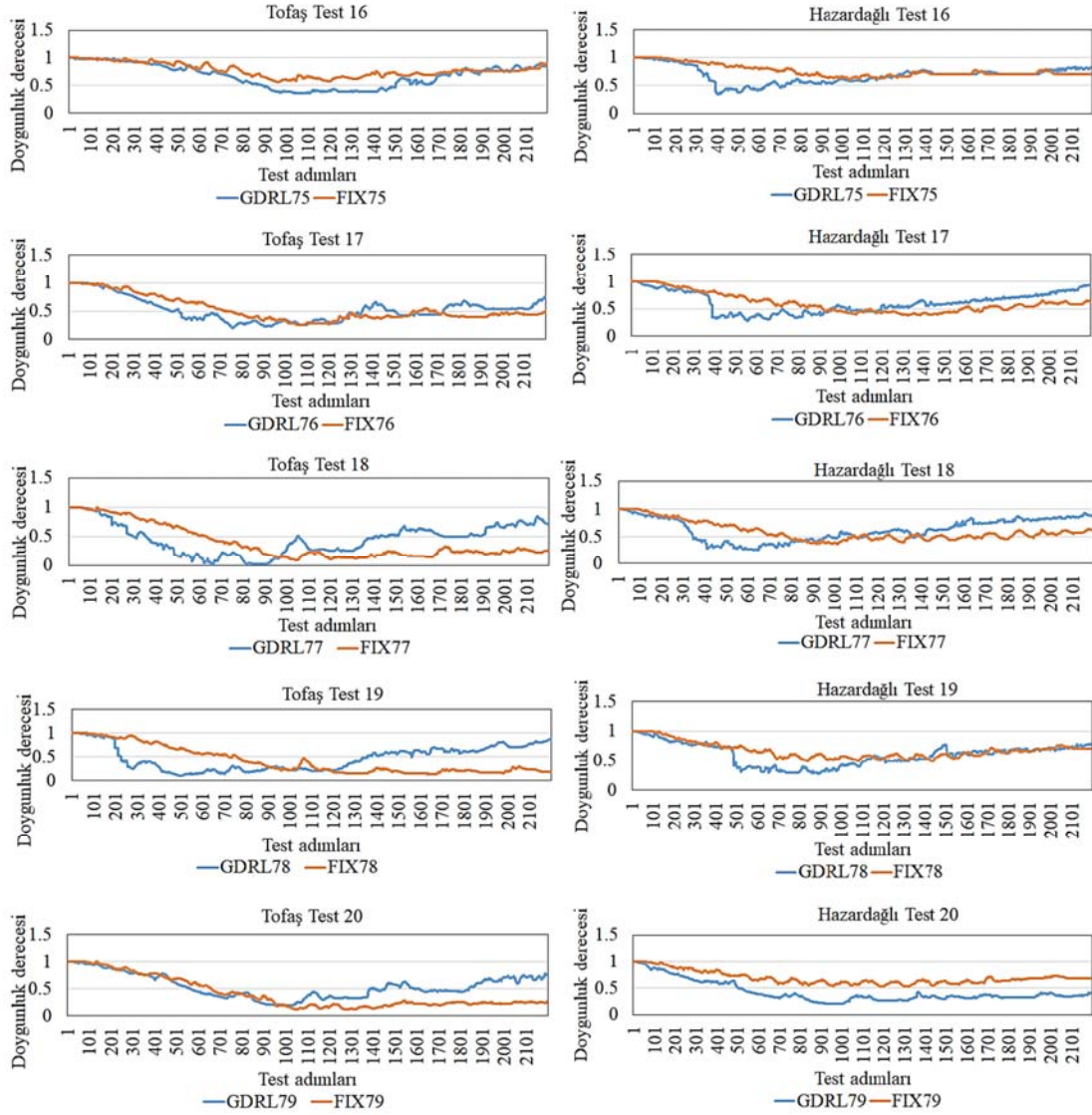
Şekil 19. GDRL eğitim kavşak ödül değerleri (GDRL training intersection award values)



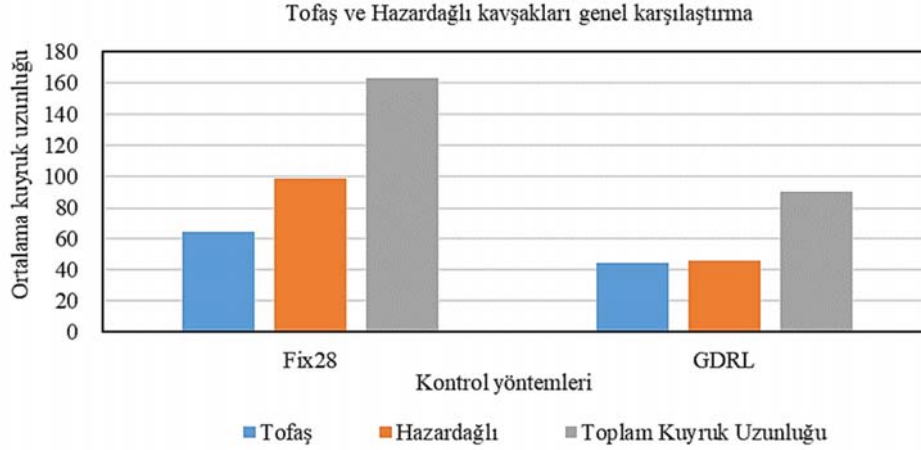
Şekil 20. Tofaş kavşağı test grafiği (Tofaş intersection test chart)



Şekil 21. Hazardağlı kavşağı test grafiği (Hazardağlı intersection test chart)



Şekil 22. Tofaş ve Hazardağlı kavşaklarının doymunluk dereceleri (Saturation degrees of Tofaş and Hazardağlı intersections)



**Şekil 23.** Genel kavşak ortalamaları (General intersection averages)

**Tablo 3.** GDRL algoritmasının karşılaştırmalı performans verileri (Comparative performance data of the GDRL algorithm)

Referans	Bağlama Hedefi	Parametre	Veri Tipi	Kavşak	Kontrol Metodu	Simülasyon Verimlilik
[44]	Ortalama seyahat süresini en aza indirme	Kuyruk uzunluğu	Gerçek veri	Ağ	RL, Markov	CityFlow %24
[42]	Birim zaman maksimum araç geçiş sayısı	Kuyruk uzunluğu	Gerçek veri	Ağ	Faz DRL	SUMO %12
[25]	Kuyruk uzunluklarını en aza indirme	Kuyruk uzunluğu	Gerçek veri	Ağ	Faz DRL	SUMO %23,5
[45]	Ortalama gecikmeyi azaltma	Kuyruk uzunluğu	Gerçek veri	Tekli Kavşak	Faz DRL	SUMO %14
[43]	Ortalama seyahat süresi	Kuyruk uzunluğu	Gerçek veri	Ağ	RL, LSTM	SUMO %36
[27]	Ortalama gecikmeyi azaltma	Kuyruk uzunluğu	Gerçek veri	Ağ	Faz DRL, Graf	CityFlow %30
[9]	Ortalama gecikmeyi azaltma	Kuyruk uzunluğu	Gerçek veri	Ağ	Faz DRL, Max Basınç	CityFlow %30
[41]	Ortalama gecikmeyi azaltma	Kuyruk uzunluğu	Gerçek veri	Ağ	Faz DRL, Fonskiyon Süre	CityFlow %39
GDRL	Toplam gecikmeyi en aza indirme	Kuyruk uzunluğu	Gerçek veri	Ağ	Faz DRL, Ford Fulkerson	SUMO %45

GDRL yönteminin daha az kuyruk sayısı sağlaması uygunluk derecesinin daha fazla olmasından kaynaklanır. Tofaş kavşağında test 75'te sabit süre yöntemi GDRL'ye göre daha verimli akış sağladığı, diğerlerinde ise daha verimsiz akış sağladığı görülmektedir. Bu durum Şekil 22'de görüldüğü gibi sadece Test 75'te sabit sürenin uygunluk akışı GDRL'den büyüktür. Diğer testlerde GDRL'nin uygunluk akışı genel olarak sabit süre yönteminden fazladır. Şekil 20 ve Şekil 21'de verildiği gibi GDRL yönteminin FIX28 yöntemine göre daha verimli olduğu görülmektedir. Simülasyon test sonuçlarının genel kavşak kuyruk uzunluğu ortalamalarını veren grafik Şekil 23'te verilmiştir.

Şekil 23'te araçların kavşaklardaki ortalama kuyruk uzunluklarına göre Tofaş kavşağında, GDRL yöntemi FIX28 yöntemine göre %31, Hazardağı kavşağında %54 ve genel simülasyon ağında ise %45'lik verimlilik sağlamıştır. GDRL yönteminin literatürdeki DRL ve RL çalışmalarına göre verimlilik karşılaştırması Tablo 3'te verilmiştir.

#### 4. Sonuçlar (Conclusions)

Trafik sinyalizasyon sistemlerinde yaygın olarak kullanılan RL ve DRL yöntemlerinde, sadece faz sıra ayarlaması yapan modeller genelleme performansı açısından verimli değildir. Bu yöntemler sabit faz sürelerinin kullanımı nedeniyle yerel maksimum ve yerel minimuma takılı kalma eğilimi gösterirler. Trafik sinyalizasyonunda

genel optimizasyon yapabilmek için DRL ve Graf tabanlı Ford Fulkerson yaklaşımı birleştirilerek GDRL adlı yeni bir sinyalizasyon yöntemi geliştirilmiştir. GDRL yöntemi, kavşaklardaki ortalama kuyruk uzunluklarını en aza indirebilen politika üretebilmek için tasarlanmıştır. Asimetrik trafik ağı üzerinde, çoklu kavşak yapısı kullanılarak gerçek araç verileriyle modellenmiştir. Sabit faz ve süre yöntemiyle karşılaştırıldığında araçların kavşaklardaki ortalama kuyruk uzunluklarını %45 oranında azalttığı görülmektedir. Çalışma kapsamında her kavşak için paralel çalışan ayrı GDRL modeli üretilmiştir. Paralel GDRL uygulaması her kavşağın, ağ içinde kendi verisine göre kendi modelini daha iyi geliştirme imkanı sunar. Test sonuçlarına göre GDRL modeli, gerçek zamanlı sinyalizasyon uygulamaları için umut vadetmektedir.

#### Teşekkür (Acknowledgement)

Bu çalışma Fırat Üniversitesi Bilimsel Araştırma Projeleri Koordinasyon Birimi (FÜBAP) tarafından ADEP.23.20 proje numarası ile desteklenmektedir.

#### Kaynaklar (References)

1. Dey S., Dhal G. C., Materials progress in the control of CO and CO2 emission at ambient conditions: An overview, Mater Sci Energy Technol, 2 (3), 607–623, 2019.



2. Schrank D., Tti's 2012 Urban Mobility Report, 2012.
3. Schrank D., Eisele B., Lomax T., and Bak J., Urban mobility, 2015.
4. Inrix, Congestion Costs Each American Nearly 100 hours, 2020.
5. Wei H., Zheng G., Gayah V., and Li Z., A Survey on Traffic Signal Control Methods, 1 (1), 2019.
6. Qadri M., Gökçe M.A., Öner E., State-of-art review of traffic signal control methods: challenges and opportunities, European Transport Research Review, 12 (1), 55, 2020.
7. AlZubi A.A., Alarifi A., Al-Maitah M., Alheyasat O., Multi-sensor information fusion for Internet of Things assisted automated guided vehicles in smart city, Sustain Cities Soc, 64, 2021.
8. Jadhao N. S., Jadhao A.S., Traffic Signal Control Using Reinforcement Learning, 2014 Fourth International Conference on Communication Systems and Network Technologies, 1130–1135, 2014.
9. Wei H., PressLight: Learning Max Pressure Control to Coordinate Traffic Signals in Arterial Network, KDD '19: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 1290-1298, July 2019.
10. Rasheed F., Yau K.-L. A., Noor R. Md., Wu C., Low Y.-C., Deep Reinforcement Learning for Traffic Signal Control: A Review, IEEE Access, 8, 208016–208044, 2020.
11. Abdullah N., Hua T. K., Using Ford-Fulkerson Algorithm and Max Flow-Min Cut Theorem to Minimize Traffic Congestion in Kota Kinabalu, Sabah. Journal of Information System and Technology Management, 2 (4), 18–34, 2017.
12. Turan E., Dandil B., Avci E., A New Graph Method Based on Deep Learning for Smart Intersections, Innovations in Smart Cities Applications, 5, 211–221, 2022.
13. Mercader P., Uwayid W., Haddad J.g, Max-pressure traffic controller based on travel times: An experimental analysis, Transportation Research Part C, 110, 275–290, 2019.
14. Vilarinho C., Tavares J.P., Rossetti R.J.F., Vilarinho C., Intelligent Traffic Lights : Green Time Period Negotiation, Transportation Research Procedia, 22 (2016), 325–334, 2017.
15. H., Yuan H., Chen, Y., Yu W., Wang C., Wang J., Gao Y., Traffic Light Optimization Based on Modified Webster Function, Journal of Advanced Transportation, 2021.
16. Abdulhai B., Pringle R., Karakoulas G., Reinforcement Learning for True Adaptive Traffic Signal Control, Journal of Transportation Engineering, 129 (3), 278–285, 2003.
17. Wiering M., Multi-agent reinforcement learning for traffic light control. ICML, 1151–1158, 2000.
18. LA P., Bhatnagar S., Reinforcement Learning With Function Approximation for Traffic Signal Control, IEEE Transactions on Intelligent Transportation Systems, 12 (2), 412–421, 2011.
19. Miletić M., Ivanjko E., Mandžuka S., Nečoska D. K., Combining Neural Gas and Reinforcement Learning for Adaptive Traffic Signal Control, 2021 International Symposium, 179–182, 2021.
20. Lawe S., Wang R., Optimization of Traffic Signals Using Deep Learning Neural Networks, Australasian Joint Conference on Artificial Intelligence, 9992, 403–415, 2016.
21. Vidali A., Crociani L., Vizzari G., Bandini S., A deep reinforcement learning approach to asset-liability management, Proceedings-2019 Brazilian Conference on Intelligent Systems, 216–221, 2019.
22. Muresan M., Fu L., Pan G., Adaptive Traffic Signal Control with Deep Reinforcement Learning An Exploratory Investigation, 97th Annual Meeting of the Transportation Research Board, 2019.
23. Ducrocq R., Farhi N., Deep Reinforcement Q-Learning for Intelligent Traffic Signal Control with Partial Detection, 2021.
24. Koh S. S., Zhou B., Yang P., Yang Z., Fang H., Feng J., Reinforcement Learning for Vehicle Route Optimization in SUMO, 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science & Systems (HPCC/SmartCity/DSS), 1468–1473, 2018.
25. Wei H., Yao H., Zheng G., Li Z., IntelliLight: A reinforcement learning approach for intelligent traffic light control, KDD 2018 - Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2496–2505, 2018.
26. Zheng G., Xiong Y., Zang X., Feng J., Wei H., Zhang H., Li Y., Xu K., Li Z., Learning Phase Competition for Traffic Signal Control. Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 1963–1972, 2019.
27. Wei H., Xu N., Zhang H., Zheng G., Zang X., Chen C., Zhang W., Zhu Y., Xu K., Li Z, CoLight Learning network-level cooperation for traffic signal control. Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019.
28. Varaiya P., The Max-Pressure Controller for Arbitrary Networks of Signalized Intersections, Advances in Dynamic Network Modeling in Complex Transportation Systems, 27-66, 2013.
29. Kim S., Suh W., Kim J., Traffic Simulation Software: Traffic Flow Characteristics in CORSIM, 5th International Conference on Information Science and Applications, 1–3, 2014.
30. Yu M., Fan W.D, Calibration of microscopic traffic simulation models using metaheuristic algorithms, Int. J. Transp. Sci. Technol., (6-1), 63–77, 2017.
31. Guo J., Kong Y., Li Z., Huang W., Cao J., Wei Y., A model and genetic algorithm for area-wide intersection signal optimization under user equilibrium traffic, Mathematics and Computers in Simulation, 155, 92–104, 2019.
32. Behrisch M., Bieker L., Erdmann J., Krajzewicz D., SUMO – Simulation of Urban Mobility, The Third International Conference on Advances in System Simulation, 55–60, 2011.
33. Khumara M.A.D., Fauziyah L., Kristalina P., Estimation of Urban Traffic State Using Simulation of Urban Mobility(SUMO) to Optimize Intelligent Transport System in Smart City, 2018 Int. Electron. Symp. Eng. Technol. Appl. IES-ETA, 163–169, 2019.
34. Khetarpal N., Parvate K., Wu C., Kreidieh A., Vinitisky E., Bayen A. M., Flow: Deep Reinforcement Learning for Control in SUMO, SUMO 2018- Simulating Auton. Intermodal Transp. Syst., 2, 134–151, 2018.
35. Kučera T., Chochoř J., Design of the City Logistics Simulation Model Using PTV VISSIM Software, Transp. Res. Procedia, 53, 258–265, 2021.
36. Zhang H., Feng S., Liu C., Ding Y., Zhu Y., Zhou Z., Zhang W., Yu Y., Jin H., Li Z., CityFlow: A Multi-Agent Reinforcement Learning Environment for Large Scale City Traffic Scenario, International World Wide Web Conference Committee, 2019.
37. Bautista P.B., Aguiar L.U., Igartua M.A., How does the traffic behavior change by using SUMO traffic generation tools, Computer Communications., 181 (c), 1–13, 2022.
38. Krajzewicz D., Erdmann J., Behrisch M., Bieker L., Recent Development and Applications of SUMO-Simulation of Urban MObility, Int. J. Adv. Syst. Meas., (3), 128–138, 2012.
39. Riedel T., Brunner U., Traffic Control Using Graph Theory, IFAC Proc., 26 (2), 131–134, 1993.
40. Ngaosai A., Chawachat J., Traffic Signal Management using Maximum Flow Approach for Consecutive Intersections, 2018 15th Int. Conf. Electr. Eng. Comput. Telecommun. Inf. Technol., 457–460, 2018.
41. Zhao W., Ye Y., Ding J., Wang T., Wei T., Chen M., IPDALight: Intensity- and phase duration-aware traffic signal control based on reinforcement learning, Journal of Systems Architecture, 123, 2021.
42. Li Z., Xu C., Zhang G., A Deep Reinforcement Learning Approach for Traffic Signal Control Optimization, 2021.
43. Wang Y., Xu T., Niu X., Tan C., Chen E., Xiong H., STMARL: A Spatio-Temporal Multi-Agent Reinforcement Learning Approach for Cooperative Traffic Light Control, IEEE Transactions on Mobile Computing, 2019.
44. Kanis S., Samson L., Bloembergen D., Bakker T., Back to Basics: Deep Reinforcement Learning in Traffic Signal Control, 10th Intl. Workshop on Urban Computing at ACM SIGSPATIAL, 2021.
45. Qi R., Huang J., Li H., Tan Q., Huang L., Cui J., Random Ensemble Reinforcement Learning for Traffic Signal Control, 2022.
46. Wegener A., Piorkowski M., Raya M., Hellbrück H., Fischer S., Hubaux J.-P., TraCI: An Interface for Coupling Road Traffic and Network Simulators, Proceedings of the 11th Communications and Networking Simulation Symposium, 155-163, 2008.
47. Olaverri-Monreal C., Errea-Moreno J., Diaz-Álvarez A., Biurrun-Quel C., Serrano-Arriezu L., Kuba M., Connection of the SUMO Microscopic Traffic Simulator and the Unity 3D Game Engine to Evaluate V2X Communication-Based Systems. Sensors, 18 (12), 2018.
48. Putra S., The Correction Value of Passenger-Car Equivalents for Motorcycle and Its Impact to Road Performance in Developing Countries, Procedia - Social and Behavioral Sciences, 16, 400–408, 2011.
49. Metlek S., Kayaalp K., Detection of bee diseases with a hybrid deep learning method, Journal of the Faculty of Engineering and Architecture of Gazi University, 36 (3), 1715-1732, 2021.
50. Şafak E., Doğru İ., Barışçı N., Toklu S., Internet of things based mobile driver fatigue detection using deep learning, Journal of the Faculty of Engineering and Architecture of Gazi University, 37 (4), 1869-1882, 2022.

